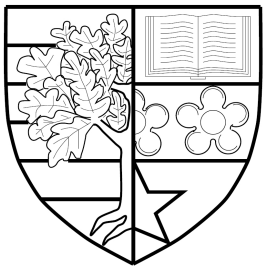
NEW SAMPLING AND OPTIMIZATION METHODS FOR TOPIC INFERENCE AND TEXT CLASSIFICATION

by

Osama Khalifa



Submitted for the degree of
Doctor of Philosophy

Department of Computer Science
School of Mathematical and Computer Sciences
Heriot-Watt University

November 2018

Abstract

Topic modelling (TM) methods, such as latent Dirichlet allocation (LDA), are advanced statistical models which are used to uncover hidden thematic structures or topics in the unstructured text. In this context, a topic is a distribution over words, and a document is a distribution over topics. Topic models are usually unsupervised; however, supervised variants have been proposed, such as supervised LDA (SLDA) which can be used for text classification. To evaluate a supervised topic model, one could measure its classification accuracy. However, unsupervised topic model's evaluation is not straightforward, and it is usually done by calculating metrics known as held-out perplexity and coherence. Held-out perplexity evaluates the model's ability to generalize to unseen documents; coherence calculates a semantic distance between the words within each topic.

This thesis explores ideas for enhancing the performance of TM, both supervised and unsupervised. Firstly, multi-objective topic modelling (MOEA-TM) is proposed, which uses a multi-objective evolutionary algorithm (MOEA) to optimize two objectives: coverage and coherence. MOEA-TM has two settings: 'start from scratch' and 'start from an estimated topic model'. In the later, the held-out perplexity is added as another objective. In both settings, MOEA-TM achieves highly coherent topics. Further, a genetic algorithm is developed with LDA log-likelihood as a fitness function. This algorithm can improve log-likelihood by up to 10%; however, perplexity scores slightly deteriorate due to over-fitting.

Hyperparameters play a significant role in TM; thus, Gibbs-Newton (GN), which is an efficient approach to learn a multivariate Pólya distribution parameter, is proposed. A closer look at the LDA model reveals that it comprises two multivariate Pólya distributions: one is used to model topics, whereas the other is used to model topics proportions in documents. Consequently, a better approach to learn multivariate Pólya distribution parameter may enhance TM. GN is benchmarked against Minka's fixed-point iteration approach, a slice sampling technique and the moments' method. We find that GN provides the same level of accuracy as Minka's fixed-point iteration method but in less time, and with better accuracy than the other approaches.

Also, LDA-GN is proposed, which makes use of the GN method in topic modelling. This algorithm can achieve better perplexity scores than the original LDA on three corpora tested. Moreover, LDA-GN is tested on a supervised task using SLDA-GN, which is the SLDA model equipped with the GN method to learn its hyperparameters.

SLDA-GN outperforms the original SLDA, which optimizes its hyperparameters using Minka's fixed point iteration method. Furthermore, LDA-GN is evaluated on a spam filtering task using the Multi-corpus LDA (MC-LDA) model; where LDA-GN shows a more stable performance compared with the standard LDA.

Finally, most topic models are based on the "Bag of Words" assumption, where a document word order is lost, and only frequency is preserved. We propose LDA-crr model, which represents word order as an observed variable. LDA-crr introduces only minor additional complexity to TM; thus, it can be applied readily to large corpora. LDA-crr is benchmarked against the original LDA using fixed hyperparameters to isolate their influence. LDA-crr outperforms LDA in terms of perplexity and shows slightly more coherent topics when the number of topics increases. Also,

LDA-crr is equipped with both the GN approach and the slice sampling technique in LDA-crrGN and LDA-crrGSS models respectively. LDA-crrGN shows a slightly better ability to generalize to unseen documents compared with LDA-GN on one corpus when the number of topics is high. However, in general, LDA-crrGSS shows better coherence scores compared with the LDA-GN and the original LDA. Furthermore, experiments to investigate LDA-crr performance in a classification task were run; thus, SLDA is extended to incorporate word orders in the SLDA-crr model. The GN and the GSS techniques are used in the SLDA-crrGN and the SLDA-crrGSS models respectively to learn its parameters. Compared with the SLDA-GN and the original SLDA, the SLDA-crrGN shows better accuracy results in classifying unseen documents. This reveals that SLDA-crrGN can pick up more useful information from the training corpus which consequently helps the model to perform better.

To my beloved family: my parents, wife, brothers and sisters.

Acknowledgements

My deepest gratitude goes to my primary supervisor Prof. David Corne for all support and guidance he gave me to complete this work. I greatly appreciate his patience, continues support, helpful and thoughtful comments, and the trust he granted me. In fact, he helped me not only in completing this thesis, but also in tracking my first steps in my future professional career. I am extremely grateful to him. Special thanks go to my second supervisor Prof. Mike Chantler for help and support he has offered throughout my course of research.

Many thanks go to all members of staff and postgraduate students in the school of Mathematical and Computer Sciences who provided help when I needed it.

This thesis would never have been possible without the support of my family: my parents, brothers, lovely sisters, and my beloved wife; their love cannot be repaid indeed. I would like to express my deep appreciation to my lovely friends who have been to me like a second family. They helped me to overcome life difficulties and they have never let me down. I am very lucky to have them all in my life.

ACADEMIC REGISTRY

Research Thesis Submission

Name*:*			
School:			
Version: *(i.e. First, Resubmission, Final)*		Degree Sought:	

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6) I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:*		Date:	

Submission

Submitted By *(name in capitals):*	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals):*			
Method of Submission *(Handed in to SSC; posted through internal/external mail):*			
*E-thesis Submitted (**mandatory for final theses)***			
Signature:		Date:	

Please note this form should be bound into the submitted thesis.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Text Mining

Large text corpora are increasingly abundant as a result of ever-speedier computational processing capabilities and ever-cheaper means of data storage. The availability of such data has encouraged research into areas such as the analysis of global events [4, 134], the measurement of consumer preferences [122], and public opinion [119]. However, the vast majority of text data available on the Internet is in unstructured formats. This has led to an increased interest in text mining and the automated extraction of useful information from such unstructured data, and particularly in the task of automated characterization and/or summarization of each document in a corpus, as well as the corpus as a whole. In Text Mining, text analyzing methods fall into two main categories:

- Statistical Methods: These rely on mathematical structures by which text can be represented. Usually, they use a "Bag of Words" principle which represents a document by a collection of words, ignoring grammar and word order but keeping frequency. Although these methods ignore essential information in the input text, research shows good and competitive results in a variety of applications.

- Linguistic Methods: These methods "understand" text by recognizing elements of the sentence. They can enable a variety of text processing applica-

tions by preserving text semantics and transforming knowledge into a machine-understandable representation.

1.2 Topic Modelling

It is generally tacitly understood that the first step in characterizing or describing an individual document is to identify the topics that are covered in that document. Thus, there is much current research into topic modelling methods such as in [68, 18, 59, 23] as algorithms that extract structured semantic topics from a collection of documents. Current topic modelling methods tend to use probabilistic models, involving many observed and hidden variables which need to be learnt from training data. They represent each topic as a distribution over corpus terms, which are a list of unique words used in the corpus. Each document in the corpus, which comprises a list of instances of corpus terms, can be represented eventually by a mixture of proportions of these topics. Probabilistic topic models such as latent Dirichlet allocation (LDA) [18] achieve this goal by clustering documents' words into topics. As a result, topic models decompose documents to a small set of topics; thus, they allow humans to gain a high-level understanding of a corpus which may be too large to be read manually. This has led researchers to use topic models as relevant tools to visualize large corpora [29, 24, 138, 109].

In addition to analyzing and visualizing large corpora, topic models have many applications in various fields such as:

- Analyzing genetics data [125, 55, 87].
- Image analysis: scene categorization [47], matching words and pictures [12], and objects discovery/tracking [132, 93].
- Studying research trends over time [63, 153] and organizing scientific research grants [146].
- Survey data processing [46].
- Social media analysis [2] and identifying health issues such as depression [128].

- Summarizing medical records [34].

1.3 Motivation and Research Gap

Topic modelling is an interesting and relatively new research area which has relevance in many applications. It becomes more and more important as the volumes of unstructured data are increasing on the World Wide Web and in institutional data repositories. These unstructured data volumes contain useful information, and it might be infeasible to process these data by humans manually. Thus, topic modelling can be used to automate the process and then to store original data in a more useful structured or semi-structured format.

Because of its unsupervised nature, topic model evaluation is relatively difficult; moreover, different applications may have particular requirements which make evaluation even harder. This leads to open problems in the topic modelling field which in turn lead to new opportunities to address these problems. One opportunity can be offered by using multi-objective evolutionary algorithms (MOEAs) in topic modelling. The MOEA approach to solving a problem depends on the problem representation on the one hand, and on the other hand, the way the problem is formulated in terms of its optimization objectives. In particular, MOEA topic modelling provides flexibility in defining the objectives that should be optimized, which may assist in finding the most suitable models that satisfy predefined requirements. To the best of our knowledge, there were no attempts in the literature with multi-objective topic models at the time of this research.

In addition, popular topic models are generally probabilistic models which, unfortunately, are intractable to be calculated exactly. Thus, approximation techniques are used in the topic modelling inference process where hyperparameters play a large role. Consequently, there is an opportunity to enhance the topic model's quality by providing efficient techniques to optimize hyperparameters values. Although there is plenty of evidence in the literature about the important role of hyperparameters in topic models [9, 151, 72], there is not enough work on investigating the best settings of these values. In addition, most authors tend to use fixed values of these

hyperparameters; therefore, they are not revealing the full potential of topic models. It is tempting to provide more optimization and sampling techniques for updating and learning hyperparameters' values automatically. To the best of our knowledge, there was no benchmark of multiple techniques on hyperparameters estimation, at the time of the research. Particularly, there is no comparison available between optimization and sampling techniques for LDA hyperparameters estimation.

Furthermore, topic models typically use the "Bag of Words" assumption which ignores grammar and word order in the input text; this might adversely affect the quality of resulted topic models. To overcome this limitation some research has already been done to relax the "Bag of Words" assumption which leads to higher quality topic models [149, 60]. However, the variable space is increased significantly, which limits the applicability of such models on large corpora. It is tempting to design topic models which relax this assumption and keep the model as simple as possible with a minimal added complexity to the base model. Providing a simple, LDA based, topic model which incorporates word order would automatically be directly applicable in most topic modelling applications including historical documents, understanding the scientific publications, computational social science, fiction and literature. To the best of our knowledge, there is no simple model with such characteristics.

1.4 Contributions

The main contributions of this thesis are summarized as follows:

- A novel multi-objective topic modelling method (MOEA-TM) in section 3.2 which could either be started from scratch or initialized by an already trained LDA model. Stand-alone and LDA initialized MOEA-TM models are evaluated against the original LDA in section 3.3. The evaluation shows that MOEA-TM is particularly useful in producing highly coherent topics. MOEA-TM is the first multi-objective topic model provided in the literature, to the best of our knowledge.

- A genetic algorithm (LDA-GA) in section 3.4, which can optimize the LDA model's log-likelihood quickly. It shows that—contrary to current thought—LDA model's log-likelihood is not necessarily correlated with a better ability to generalize unseen documents.

- Two novel methods to learn multivariate Pólya distribution parameters: firstly, the (GN) algorithm, which is based on Gibbs sampling and Newton's method, is described in section 4.3.1.1 whereas the second method, which is based on a slice sampling approach, is provided in section 4.3.1.2. These two approaches are evaluated against popular methods available in the literature in section 4.3.2. The evaluation shows that the GN approach provides the same level of accuracy as popular methods in the literature with less resource usage.

- A new model (LDA-GN) in section 4.4, which is an extension of LDA using the GN algorithm, is developed and compared with an LDA extension which uses slice sampling and with the original LDA which uses Minka's fixed point iteration method in section 4.4.3. LDA-GN shows good perplexity scores when it is compared with these other models.

- A supervised extension for LDA-GN, which is used to measure classification performance, shows a better classification performance compared with the original SLDA in section 4.4.3.2.

- LDA-crr, a novel LDA extension in section 5.4, incorporates corpus word order into a topic model without adding a large number of latent variables to the original LDA model. The new model goes beyond the "Bag of Words" assumption by adding observed variables to hold word order information. It is benchmarked against the original LDA and the LDA-GN in section 5.6. Generally speaking, LDA-crr shows a better ability to generalize to held-out documents and to produce more coherent topics.

- A supervised extension for LDA-crr, which is provided in section 5.5, incorporates word sequence order in the modelling process. This new model performs

better than the original LDA and LDA-GN in classification tasks, as is shown in section 5.7

1.5 Overview

The reminder of this thesis is organized as follows:

Chapter 2 provides a background on basic topic models and multi-objective optimization problems. It starts with presenting a concise review of probabilistic distributions mainly used in topic models. Then it illustrates basic unsupervised and supervised topic models, followed by a detailed description of the two most popular approaches to estimate topic models, which are: variational inference and Gibbs sampling. All models in this thesis are implemented using the Gibbs sampling technique; hence, the variational inference is illustrated only to contrast it with Gibbs sampling and is not needed to understand this thesis. Next, it highlights popular methods which are available in the literature to evaluate topic models. Eventually, it presents a background on multi-objective optimization and details on the MOEA/D framework.

Chapter 3 presents a novel multi-objective topic modelling algorithm (MOEA-TM) which uses MOEA/D to optimize both topic coherence and the coverage of training documents. Later, it benchmarks this model against the original LDA to measure its performance; unfortunately, MOEA-TM is not able to optimize the ability to generalize to unseen documents which consequently limits its applications. Eventually, it illustrates a novel genetic algorithm based on the LDA model to optimize the model's log-likelihood. Although it can optimize the model's log-likelihood, the model's ability to generalize to unseen documents is deteriorated which limits the use of genetic algorithm optimization techniques in topic models.

Chapter 4 introduces two novel methods to learn multivariate Pólya distribution parameters from data samples. The first method uses Gibbs sampling and Newton optimization techniques, whereas the second method uses a slice sampling approach. Then, it benchmarks these methods against popular techniques in terms of accuracy and speed. Based on these methods, Chapter 4 extends the LDA model and com-

pares it against the original LDA in order to check the new extension's performance. Eventually, it evaluates these models on a supervised task.

Chapter 5 presents a novel extension for the LDA which relaxes the "Bag of Words" assumption and incorporates word order information in a topic model. The new model is evaluated against other models using multiple metrics to show its performance. Eventually, Chapter 6 concludes this thesis by providing a summary of key research findings and future work ideas in topic modelling, and how this research can be extended.

Chapter 2

Background on Topic Models and Multi-Objective Optimization

This chapter covers the important concepts behind topic modelling. Firstly, it provides a brief background on the basic distributions used in typical topic models. After that, a background on topic modelling is presented which covers both supervised and unsupervised models. Supervised topic models are used in this thesis as another means for evaluation and to check performance in a supervised task such as classification. Then it describes well-known techniques to estimate topic models because most interesting topic models are intractable and need to be approximated. After that, it illustrates topic modelling evaluation techniques that are used in this thesis. Eventually, it provides a background on multi-objective optimization problems, which is used in Chapter 3 where multi-objective optimization is employed in topic modelling to investigate the possibility of tuning performance.

2.1 Preliminaries: Topic Modelling Related Distributions

Topic models are statistical models to uncover hidden thematic features in a collection of documents. Thus, a dataset of text documents can be modelled as an output of a probabilistic process using combinations of probabilistic distributions. In this

section, a background on basic topic modelling probabilistic distributions used in this thesis is presented.

2.1.1 Multinomial Distribution

The multinomial distribution is a discrete distribution to model the output counts of rolling a K-sided biased die N times. Let $X = (X_1, X_2, .., X_K)$ be a random variable where each component X_i represents the number of times side i appears, and let $\rho = (\rho_1, \rho_2, .., \rho_K)$ be a vector to represent the probabilities of each side of the die. The two variables X and ρ should satisfy the following conditions: $\sum_{i=1}^{K} X_i = N$ and $\sum_{i=1}^{K} \rho_i = 1$. Consequently, the probability of getting the variable X is given by the following formula:

$$P(X; \rho) = \frac{N!}{\prod_{i=1}^{K} X_i!} \prod_{i=1}^{K} \rho_i^{X_i} \quad . \tag{2.1}$$

The multinomial distribution is a common choice to model terms in the text mining area [96]. It can be considered as a unigram language model to calculate the probability of a group of words or a document.

2.1.2 Dirichlet Distribution

The Dirichlet distribution is a distribution over a $K-1$ dimensional probability simplex in a K dimensional space. It is a multivariate generalization of the beta distribution. Consider a bag of infinite K sided biased dice; each die is unfair in a different way and it has its own probability mass function (PMF). The Dirichlet distribution can be used to model the randomness of these PMFs. Thus, let $Q = (Q_1, Q_2, .., Q_K)$ be a random variable where each component Q_i is a positive number and $\sum_{i=1}^{K} Q_i = 1$; consequently, Q represents a $K-1$ dimensional simplex. Let $\alpha = (\alpha_1, \alpha_2, .., \alpha_K)$ where each component $\alpha_i > 0$. Then, Q is distributed by Dirichlet with a parameter α if it has the following density function:

$$P(Q; \alpha) = \frac{\prod_{i=1}^{K} Q_i^{\alpha_i - 1}}{B(\alpha)} \quad . \tag{2.2}$$

In this equation, $B(\alpha)$ is the Dirichlet distribution's normalization constant, which is a multivariate generalization of the beta function, the normalizing constant of the beta distribution. The Dirichlet distribution's normalization constant function is given by the following formula:

$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \ , \tag{2.3}$$

where, $\Gamma(\alpha_i)$ is the gamma function [36]. One interesting aspect of the Dirichlet distribution is that when the values α_i are less than 1, points on the edges of the simplex get higher probabilities than points in the middle. In the context of topic modelling, this enables a topic model to generate a more distinct topics set.

2.1.3 Multivariate Pólya Distribution

The Multivariate Pólya distribution, also known as the Dirichlet-Multinomial distribution, is a compound distribution. Sampling from a multivariate Pólya distribution involves sampling a vector ρ from a K dimensional Dirichlet distribution with a parameter α and then drawing a set of discrete samples from a categorical distribution with parameter ρ. This process corresponds to the 'Pólya urn' which comprises sampling with replacement from an urn containing coloured balls. Every time a ball is sampled, its colour is observed and it is replaced into the urn; then an additional ball with the same colour is added to the urn.

Consider a K dimensional data-count observation vector π to be generated using Dirichlet and Multinomial distributions in the following procedure:

1. Draw a proportion ρ from Dirichlet distribution: $\rho \sim Dir(\alpha)$

2. Draw N IID samples π_i from Multinomial [1]: $\pi_i \sim Mult(\rho)$; then generate counts vector using $\pi = \sum_{i=1}^{N} \pi_i$.

[1] Multinomial distribution with trials number equal to one is used to generate each sample. Following this procedure the joint probabilities for all generated samples is the same as the categorical distribution because Multinomial distribution constant is reduced to one [107].

Consequently, the resulting joint probability is given by the following formula:

$$P(\pi, \rho; \alpha) = \frac{\Gamma(\alpha_\circ)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \rho_k^{\pi_k + \alpha_k - 1} \quad , \tag{2.4}$$

where $\alpha_\circ = \sum_{k=1}^{K} \alpha_k$ and $\pi_\circ = \sum_{k=1}^{K} \pi_k = N$. In addition, the resulting data-count vector π is distributed under multivariate Pólya distribution with parameter α, which is defined by the marginal distribution $P(\pi; \alpha)$ as follows:

$$\begin{aligned} P(\pi; \alpha) &= \int_\rho P(\rho; \alpha) P(\pi|\rho)\, d\rho \\ &= \frac{B(\pi + \alpha)}{B(\alpha)} \quad , \end{aligned} \tag{2.5}$$

where B is the Dirichlet normalisation constant function defined in Equation 2.3. In Bayesian modelling, $P(\rho; \alpha)$ is called the prior distribution; whereas, $P(\rho|\pi, \alpha)$ is called the posterior distribution. The prior can be considered as a previous belief before the data is seen; on the other hand, the posterior reflects both the prior belief and the observed data. For the multivariate Pólya distribution, the posterior can be calculated from the prior and marginal distributions as follows:

$$\begin{aligned} P(\rho|\pi, \alpha) = \frac{P(\rho, \pi; \alpha)}{P(\pi; \alpha)} &= \frac{1}{B(\pi + \alpha)} \prod_{k=1}^{K} \rho^{\pi_k + \alpha_k - 1} \\ &= Dir(\pi + \alpha) \quad . \end{aligned} \tag{2.6}$$

Because the resulting posterior distribution is from the same family as the prior distribution, the prior distribution is called 'conjugate prior' [126] to the multinomial distribution likelihood. This feature is used in topic models to simplify the inference process, as shown in the following section.

2.2 Topic Modelling

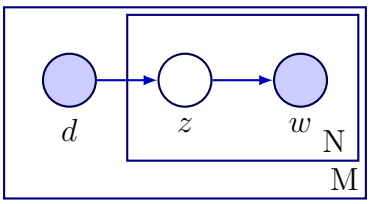
Topic modelling is a technique to analyse large amounts of unclassified text data [144]. It exploits the statistical regularities that occur in natural language documents to match queries to documents in a way that, though entirely statistical, carries

strong semantic resonance. On the one hand, good topic models should deal with synonymy, i.e. connect words with similar meanings which typically co-occur within topics. On the other hand, it should be able to distinguish polysemy [49] where words can have multiple meanings depending on context (e.g. the word 'set' will appear with high probability in both a 'tennis' topic and a 'discrete mathematics' topic). Eventually, a document can be described as a distribution over topics which are themselves distributions over corpus terms. In this thesis, the word 'term' is used to describe a unique word in the whole corpus, whereas 'word' refers merely to a word from a corpus, i.e. a single instance of a term.

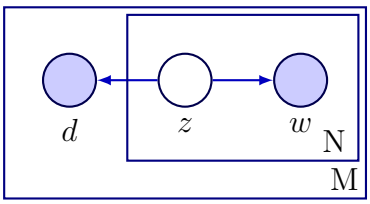
Let $W = [W_1, W_2, .., W_M]$ be a corpus with M documents. Each document comprises a collection of words $W_d = [W_d^1, W_d^2, .., W_d^{N_d}]$ where N_d is number of words in document W_d. Most current topic models use the "Bag of Words" (BoW) assumption [66], where the order of the words in the documents is lost and only their frequencies are preserved. BoW simplifies the input of topic models and consequently, allows us to design simpler models which give relatively good results without consuming a lot of resources. Let $w = [w_1, w_2, .., w_V]$ be unique words or total terms in the whole corpus and let V be the total number of terms. Then, a topic φ_i is a discrete distribution over the V corpus terms. Given the words of the documents as an observed variable, the topic model's objective is to estimate the topics φ and their proportions in corpus documents θ. For interesting statistical models such as LDA, the exact calculation of φ and θ is intractable even for small corpora; thus, approximation techniques are used for this purpose. In addition, Topic modelling is a multimodal and non-concave problem [129] which makes learning topic model variables not an easy task.

2.2.1 Unsupervised Probabilistic Topic modelling

Most topic models are unsupervised learning tools [17], which start from documents' words as the only observed variable to learn φ the topics and θ their proportions in corpus documents. Consequently, topic modelling can be used to organise and give insights to help understand unstructured data. An early topic model is the latent



(a) Asymmetric Parametrization



(b) Symmetric Parametrization

Figure 2.1: pLSA Plate Model

semantic analysis (LSA) [38], which formed the basis of all other topic models in spite of the fact that it is not probabilistic. One year later, the probabilistic latent semantic analysis (pLSA) approach [68] was provided. Based on pLSA, Latent Dirichlet Allocation (LDA) [18] was proposed as a standard and now highly popular topic modelling approach.

2.2.1.1 Probabilistic Latent Semantic Analysis (pLSA)

pLSA [68] is a probabilistic latent variable model for co-occurrence analysis. This model is based on the BoW assumption; thus, all corpus documents are represented by an $M \times V$ matrix X. where, rows represent document, columns represent terms and each entry in X represents number of times a term $w \in [1, .., V]$ occurs in a document $d \in [1, ..M]$. Moreover, a latent class variable $z \in [1, .., K]$ is used to model topic assignments; where K is a predefined total number of topics.

Figure 2.1 shows a plate notation graphical representation for pLSA model with symmetric and asymmetric parametrization. Algorithm 1 defines the pLSA symmetric parametrization generative process for the document-term matrix X. Consequently, the joint probability distribution for one co-occurrence in X is given by:

Algorithm 1 pLSA generative process

for $d = 1$ **to** M **do**
  for $n = 1$ **to** N_m **do**
    Draw a topic $k \sim P(z),\ k \in 1..K$
    Draw a document $d \sim P(d|z = k)$
    Draw a word $w \sim P(w|z = k)$
    $X_{d,w} \leftarrow X_{d,w} + 1$
  end for
end for

$$P(d,w) = \sum_{z=1}^{K} P(z)P(d|z)P(w|z) \tag{2.7}$$

Whereas, the probability distribution over X is given by:

$$P(X) = \prod_{d=1}^{M}\prod_{w=1}^{V}\left(\sum_{z=1}^{K} P(z)P(d|z)P(w|z)\right)^{X_{d,w}} \tag{2.8}$$

Thus, the pLSA model parameters are $\mathcal{M} = \{P(z), P(d|z), P(w|z)\}$; where, the values $P(w|z)$ represents a $K \times V$ scalar variables which can be used to define topic distributions over terms, whereas $P(d|z)$ comprises $K \times M$ scalar variables which can be used to calculate topic mixtures in corpus documents. As a result, the pLSA model comprises $K \times (V + M)$ parameters which need to be estimated based on observed documents-terms co-occurrences.

To learn these parameters, the standard expectation maximization (EM) technique [40] can be used. EM is a technique to find parameter estimations which maximize the likelihood of the model, by alternating between two steps: the expectation (E) step, where likelihood is calculated using current estimates of parameters, and the maximization (M) step, which is used to get a more accurate estimation of the parameters based on the current expectation. Consequently, the pLSA (E) step equation is given by:

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')} \tag{2.9}$$

Whereas the (M) step equations [69] are as follows:

$$\begin{aligned} P(z) &\propto \sum_{d=1}^{M}\sum_{w=1}^{V} X_{d,w}P(z|d,w) \\ P(d|z) &\propto \sum_{w=1}^{V} X_{d,w}P(z|d,w) \\ P(w|z) &\propto \sum_{d=1}^{M} X_{d,w}P(z|d,w) \end{aligned} \tag{2.10}$$

The main advantage of pLSA is that it is a simple probabilistic model which can

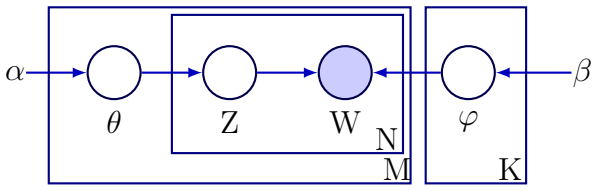


Figure 2.2: LDA model

be easily extended and embedded in other models. However, this model suffers from multiple limitations. Firstly, it can be seen from Algorithm 1 that the pLSA generative process is not well-defined and there is no natural way to generate unseen documents. Moreover, pLSA parameters increase linearly with the number of training documents which can lead to serious overfitting problems [18].

2.2.1.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) [18] is among the most prominent of current topic modelling techniques. LDA is a statistical model of document collection, which considers corpus documents to be underpinned by a mixture of latent topics, where each topic is characterized by a multinomial distribution over vocabulary words. In order to overcome pLSA limitations—its linear variable growth and its poorly defined generative process—, the Dirichlet distribution is used in LDA. Because it is a conjugate prior to the multinomial distribution, the Dirichlet distribution is a natural choice to model and generate θ and φ variables. Consequently, Dirichlet parameters α and β become the model's hyperparameters. The hyperparameter α controls the generation of topic mixtures θ, and hyperparameter β is used to control the generation of topics φ. In the LDA model, the only observed variables are the documents' words and all the rest need to be estimated.

The plate notation graphical representation of LDA in Figure. 2.2 illustrates the relationship between latent and observed variables. Meanwhile, the LDA generative process described in Algorithm 2 defines a joint probability distribution over these variables as follows:

$$P(W, Z, \theta, \varphi|\alpha, \beta) = \prod_{k=1}^{K} P(\varphi_k|\beta) \prod_{d=1}^{M} P(\theta_d|\alpha) \prod_{t=1}^{N_d} P(Z_{d,t}|\theta_d) P(W_{d,t}|\varphi_{Z_{d,t}}) \quad (2.11)$$

Algorithm 2 LDA generative process

for $k = 1$ **to** K **do**
  Draw a topic $\varphi_k \sim Dir(\beta)$
end for
for $d = 1$ **to** M **do**
  Draw a topic proportion $\theta_d \sim Dir(\alpha)$
  for $t = 1$ **to** N_d **do**
    Draw a topic $Z_{d,t} \sim Multi(\theta_d),\ Z_{d,t} \in 1..K$
    Draw a word $W_{d,t} \sim Multi(\varphi_{Z_{d,t}})$
  end for
end for

where N_d is the number of words in the document W_d. The conjugacy between Dirichlet and multinomial distributions allows θ and φ to be marginalized out:

$$P(W, Z|\alpha, \beta) = \prod_{d=1}^{M} \frac{B(\widehat{z_{d,\circ}} + \alpha)}{B(\alpha)} \cdot \prod_{k=1}^{K} \frac{B(\widehat{z_{\circ}^{k}} + \beta)}{B(\beta)} \tag{2.12}$$

where, $\widehat{z_{d,\circ}}$ is a vector of length K, and each component value $\widehat{z_{d,\circ}^{k}}$ represents number of words in document W_d assigned to the topic k. On the other hand, $\widehat{z_{\circ}^{k}}$ is a vector of length V; each component value $\widehat{z_{\circ,r}^{k}}$ represents the number of instances of term r in the whole corpus that are assigned to topic k. The key inference problem that needs to be calculated is the posterior distribution given by the formula:

$$P(Z|W, \alpha, \beta) = \frac{P(W, Z|\alpha, \beta)}{P(W|\alpha, \beta)} = \frac{\prod_{d=1}^{M} \frac{B(\widehat{z_{d,\circ}}+\alpha)}{B(\alpha)} \cdot \prod_{k=1}^{K} \frac{B(\widehat{z_{\circ}^{k}}+\beta)}{B(\beta)}}{\sum\limits_{Z} \left(\prod_{d=1}^{M} \frac{B(\widehat{z_{d,\circ}}+\alpha)}{B(\alpha)} \cdot \prod_{k=1}^{K} \frac{B(\widehat{z_{\circ}^{k}}+\beta)}{B(\beta)} \right)} \tag{2.13}$$

Unfortunately, the exact calculation of the posterior distribution is generally intractable due to the denominator. Its calculation involves summing over all possible settings of the topic assignment variable Z. This number has an exponential value given by K^N where $N = \sum_{d=1}^{M} N_d$ is the total number of corpus words; hence, LDA exact inference is an NP-hard problem [139]. However, there are several approximation algorithms to sample from the posterior distribution which can be used for LDA such as: variational inference methods [18, 67], expectation propagation [105], and Gibbs sampling [143, 59, 125]. Variational methods and Markov-chain Monte Carlo methods such as Gibbs sampling are widely used in the literature. More details

about LDA model inference are elaborated in section 2.3.

2.2.2 Supervised Probabilistic Topic Modelling

Most topic models such as LDA and many of its extensions are unsupervised, where the only observed variables are the documents' words. However, supervision can be introduced to topic models by not modelling corpus words only, but also document's labels or tags. Supervised latent Dirichlet allocation (SLDA) [17], labelled latent Dirichlet allocation (LLDA) [127], maximum entropy discrimination latent Dirichlet allocation (MedLDA) [163] are examples of commonly used models.

2.2.2.1 Supervised LDA (SLDA)

Supervised latent Dirichlet allocation (SLDA) [17] is one of the most straightforward and most commonly used supervised topic models. Its topics are not only dependent on document words but also on document label variables. Thus, in order to fully train an SLDA model, labelled documents should be provided as an input; where each training document has one class or label associated with it. Figure 2.3 shows a graphical plate representation of the SLDA model. SLDA is designed as an extension to LDA for classification tasks [28], where a response variable associated with each document is added to model documents' labels. In order to find topics

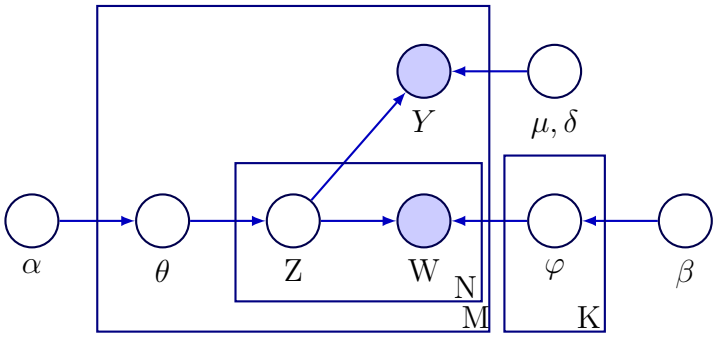


Figure 2.3: SLDA model

which best describe new data, SLDA jointly models words and response variables $Y = [Y_1, Y_2, .., Y_M]$, which is an $M \times 1$ vector, where M is total number of documents. The response variable Y is modelled under the generalized linear model (GLM) [99] [113]. GLM is a generalization of linear models in which each response variable Y_d

is assumed to be generated from an exponential family distribution with canonical parameter $\overline{Z_d} \cdot \mu$ and dispersion parameter δ. Where, μ is a $K \times 1$ vector and $\overline{Z_d}$ is a row vector which represents topics discrete distribution for document W_m and given by:

$$\overline{Z_d} = \left[\frac{\widehat{z^1_{d,\circ}}}{N_d}, \frac{\widehat{z^2_{d,\circ}}}{N_d}, .., \frac{\widehat{z^K_{d,\circ}}}{N_d} \right] \quad , \tag{2.14}$$

where, $\widehat{z^i_{d,\circ}}$ is total number of terms in document W_d assigned to topic $i \in [1..K]$. Consequently, the probability of the response variable for a given document W_d is given by the following formula:

$$P(Y_d|Z_d, \mu, \delta) = \exp\left(\frac{Y_d(\overline{Z_d} \cdot \mu) - A(\overline{Z_d} \cdot \mu)}{\delta} + G(Y_d, \delta) \right) \quad . \tag{2.15}$$

For Normal distribution, which is what is used in this thesis, the two functions A and G are given by the following formulas:

$$\begin{aligned} A(\overline{Z_d} \cdot \mu) &= \frac{\left(\overline{Z_d} \cdot \mu\right)^2}{2} \\ G(Y_d, \delta) &= -\frac{{Y_d}^2}{2\delta} + \log\left(\frac{1}{\sqrt{2\pi\delta}}\right) \quad . \end{aligned} \tag{2.16}$$

Algorithm 3 SLDA generative process

for $k = 1$ **to** K **do**
  Draw a topic $\varphi_k \sim Dir(\beta)$
end for
for $d = 1$ **to** M **do**
  Draw a topic proportion $\theta_d \sim Dir(\alpha)$
  for $n = 1$ **to** N_d **do**
    Draw a topic $Z_{d,n} \sim Multi(\theta_d),\ Z_{d,n} \in 1..K$
    Draw a word $W_{d,n} \sim Multi(\varphi_{Z_{d,n}})$
  end for
  Draw response variable $Y_d \sim GLM(\overline{Z_d}, \mu, \delta)$
end for

From its plate graphical representation shown in Figure 2.3 and its generative process illustrated in Algorithm 3, the following formula gives the joint probability for the

SLDA model:

$$P(W, Y, Z, \theta, \varphi|\alpha, \beta, \mu, \delta) = \prod_{k=1}^{K} P(\varphi_k|\beta) \cdot \prod_{d=1}^{M} P(Y_d|\overline{Z}_d, \mu, \delta) P(\theta_d|\alpha) \prod_{t=1}^{N_d} P\left(Z_{d,t}|\theta_d\right) P\left(W_{d,t}|\varphi_{Z_{d,t}}\right) \quad (2.17)$$

Thanks to the conjugacy between Dirichlet and Multinomial distributions, which enables integrating out θ and φ in a closed form easily. Thus, the resulted marginal distribution is given by the following formula:

$$P(W, Y, Z|\alpha, \beta, \mu, \delta) = \prod_{k=1}^{K} \frac{B(\widehat{z_{\circ}^{k}} + \beta)}{B(\beta)} \cdot \prod_{d=1}^{M} \frac{B(\widehat{z_{d,\circ}} + \alpha)}{B(\alpha)} P(Y_d|\overline{Z}_d, \mu, \delta) \quad (2.18)$$

The main inference problem for SLDA is to calculate the posterior distribution $P(Z|W, Y, \alpha, \beta, \mu, \delta)$, which is given by the following formula:

$$P(Z|W, Y, \alpha, \beta, \mu, \delta) = \frac{P(Z, W, Y|\alpha, \beta, \mu, \delta)}{P(W, Y|\alpha, \beta, \mu, \delta)} \quad (2.19)$$

The exact calculation of the posterior distribution is intractable because it involves summing over an exponential number of different settings of variable Z. Fortunately, the posterior can be approximated using Variational methods or Gibbs sampling techniques. The detailed inference methods of both LDA and SLDA are explained in the following section.

2.3 Probabilistic Topic Model Inference

The main problem in topic models is to calculate the posterior distribution after observing corpus words and documents. However, for interesting topic models such as LDA and its extensions, the exact calculation of the posterior is intractable [18]. Therefore, a variety of approximation techniques have been developed to solve topic modelling's main problem. For pLSA, a standard EM algorithm [40] to estimate parameters that maximize the likelihood can be used. These are the settings of

pLSA model parameters which represent the likelihood's mode.

$$\arg\max_{\theta,\varphi,Z} \mathcal{L}(\theta, \varphi, Z|W)$$

However, in the LDA θ and φ are treated as hidden variables not parameters; thus, EM can be used to compute a maximum a posteriori (MAP) estimate for the model's random variables [9]; hence, MAP estimation represents the posterior distribution mode.

$$\arg\max_{\theta,\varphi,Z} [P(\theta, \varphi, Z|W, \alpha, \beta) \propto P(W|Z, \theta, \varphi)P(\theta|\alpha)P(\varphi|\beta)]$$

In general, it is more accurate to learn more about the posterior distribution and calculate its mean, instead of learning only the mode value. Thus, many methods can be used to achieve this goal including variational Bayes and Gibbs sampling techniques.

2.3.1 Variational Bayes (VB)

The idea behind variational Bayes (VB) is to find a family of distributions $Q(x|\xi)$, with its own variational parameters ξ, to approximate the true intractable posterior $P(x|D)$. Where, D is observed data and x represents hidden variables. Consequently, Kullback-Leibler divergence (KL-divergence) [86, 85] between $Q(x|\xi)$ and the true posterior is given by the following equation [2]:

$$\begin{aligned} KL(Q(x)||P(x|D)) &= -\int Q(x) \log\left(\frac{P(x|D)}{Q(x)}\right) \mathrm{d}x \\ &= -\int Q(x) \log\left(\frac{P(x,D)}{Q(x)}\right) \mathrm{d}x + \log(P(D)) \end{aligned} \tag{2.20}$$

Maximizing the following free energy function:

$$F(Q(x)) = \int Q(x) \log\left(\frac{P(x,D)}{Q(x)}\right) \mathrm{d}x \ , \tag{2.21}$$

[2]For clarity, the notational dependence of function Q or its factors on variational parameter ξ is omitted sometimes.

is equivalent to minimizing the KL-divergence in Equation 2.20 and it leads to a tightened evidence lower bound (ELBO) function. The free energy function in Equation 2.21 can be decomposed into two functions:

$$\begin{aligned} F(Q(x)) &= \int Q(x) \log \left(P(x, D)\right) \, \mathrm{d}x - \int Q(x) \log(Q(x)) \mathrm{d}x \\ &= \mathbb{E}_Q \left[\log P(x, D)\right] + \mathbf{H}(Q(x)) \quad . \end{aligned} \tag{2.22}$$

Where, $\mathbb{E}_Q \left[\log P(x, D)\right]$ is the expected log joint and $\mathbf{H}(Q(x))$ is Shannon entropy[19]. When dealing with models with multiple variables, usually a form of VB called mean-field variational Bayes is used, in which the approximation distributions $Q(x)$ is assumed to factorise into single variable factors:

$$Q(x|\xi) = \prod_i Q_i(x_i|\xi_i) \tag{2.23}$$

It is important to emphasis here that each function $Q_i(x_i)$ is an approximate posterior for the i^{th} variable, not an approximation for a marginal. Consequently, under the mean-field assumption the free energy function can be rewritten as follows [52]:

$$\begin{aligned} F(Q(x)) &= \int \prod_i Q_i(x_i) \log P(x, D) \mathrm{d}x - \int \left(\prod_i Q_i(x_i)\right) \sum_i \log Q_i(x_i) \mathrm{d}x \\ &= -KL\left(Q_j(x_j) || \exp\left(\mathbb{E}_{Q_{\neg j}}\left[\log P(x, D)\right]\right)\right) + \mathbf{H}(Q_{\neg j}(x_{\neg j})) + C \end{aligned} \tag{2.24}$$

This is interesting because trying to minimize the KL-divergence between two large joint distributions, which is hard, ends up with minimizing the KL-divergence for much easier distributions. In the following section, a variational Bayes inference method for LDA is elaborated.

2.3.1.1 LDA Variational Bayes Inference

The first step in variational Bayes inference is to choose a good tractable distribution family Q to approximate the posterior $P(Z, \theta, \varphi|W, \alpha, \beta)$. A closer look at the LDA model reveals that the coupling between θ and φ is what makes the model intractable [42]. Consequently, a simple way to choose good Q functions is by dropping the

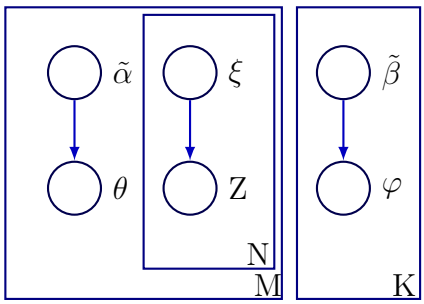


Figure 2.4: Variational distribution Q for LDA

edges which cause this problematic coupling. Figure 2.4 shows the LDA model after decoupling θ and φ and adding the variational parameters. Consequently, the distribution $Q(Z, \theta, \varphi|\tilde{\alpha}, \tilde{\beta}, \xi)$ is given by the following equation:

$$Q(Z, \theta, \varphi|\tilde{\alpha}, \tilde{\beta}, \xi) = \prod_{k=1}^{K} Q(\varphi_k|\tilde{\beta}_k) \prod_{m=1}^{M} \left(Q(\theta_m|\tilde{\alpha}_m) \prod_{n=1}^{N_m} Q(Z_{m,n}|\xi_{m,n}) \right) \quad ; \tag{2.25}$$

where,

$$\begin{aligned} Q(\varphi_k|\tilde{\beta}_k) &\sim Dir(\tilde{\beta}_k) \\ Q(\theta_m|\tilde{\alpha}_m) &\sim Dir(\tilde{\alpha}_m) \\ Q(Z_{m,n}|\xi_{m,n}) &\sim Mult(\xi_{m,n}) \quad . \end{aligned}$$

It is clear that the distribution $Q(Z, \theta, \varphi|\tilde{\alpha}, \tilde{\beta}, \xi)$ totally factorises into single variables; hence, mean-field VB can be used. Substituting variational distribution Equation 2.25 and LDA joint distribution Equation 2.11 in variational free energy Equation 2.22 gives the following ELBO function for LDA:

$$\begin{aligned} F(Q(Z, \theta, \varphi)) = \mathbb{E}_Q \left[\log P(\theta|\alpha)\right] + \mathbb{E}_Q \left[\log P(Z|\theta)\right] + \mathbb{E}_Q \left[\log P(W|\varphi, Z)\right] \\ + \mathbb{E}_Q \left[\log P(\varphi|\beta)\right] + \mathbf{H}\left(Q(\theta)\right) + \mathbf{H}\left(Q(Z)\right) + \mathbf{H}\left(Q(\varphi)\right) \end{aligned} \tag{2.26}$$

Let $\mathbb{1}$ be the indicator function, so expectations from the ELBO function are given by the following formulae:

$$\begin{aligned}
\mathbb{E}_Q[\log P(\theta|\alpha)] &= \sum_{m=1}^{M}\left(\log\Gamma(\alpha_\circ) + \sum_{k=1}^{K}(\alpha_k-1)\Big(\Psi(\tilde{\alpha}_{m,k}) - \Psi(\tilde{\alpha}_{m,\circ})\Big) - \log\Gamma(\alpha_k)\right)\\
\mathbb{E}_Q[\log P(\varphi|\beta)] &= \sum_{k=1}^{K}\left(\log\Gamma(\beta_\circ) + \sum_{v=1}^{V}(\beta_v-1)\Big(\Psi(\tilde{\beta}_{k,v}) - \Psi(\tilde{\beta}_{k,\circ})\Big) - \log\Gamma(\beta_v)\right)\\
\mathbb{E}_Q[\log P(Z|\theta)] &= \sum_{m=1}^{M}\sum_{n=1}^{N_m}\sum_{k=1}^{K}\xi_{m,n}^{k}\Big(\Psi(\tilde{\alpha}_{m,k}) - \Psi(\tilde{\alpha}_{m,\circ})\Big)\\
\mathbb{E}_Q[\log P(W|\varphi,Z)] &= \sum_{m=1}^{M}\sum_{n=1}^{N_m}\sum_{k=1}^{K}\sum_{v=1}^{V}\mathbb{1}\left(W_{m,n}=v\right)\xi_{m,n}^{k}\Big(\Psi(\tilde{\beta}_{k,v}) - \Psi(\tilde{\beta}_{k,\circ})\Big)\ ,
\end{aligned}$$

where, $\Psi(x) = \frac{\partial \log\Gamma(x)}{\partial x}$ is the digamma function the logarithmic derivative of gamma function [36]. In addition, the entropies of the function F are given by:

$$\begin{aligned}
\mathbf{H}(Q(\theta)) &= \sum_{m=1}^{M}\left(-\log\Gamma(\tilde{\alpha}_{m,\circ}) + \sum_{k=1}^{K}\log\Gamma(\tilde{\alpha}_{m,k}) - (\tilde{\alpha}_{m,k}-1)\Big(\Psi(\tilde{\alpha}_{m,k}) - \Psi(\tilde{\alpha}_{m,\circ})\Big)\right)\\
\mathbf{H}(Q(\varphi)) &= \sum_{k=1}^{K}\left(-\log\Gamma(\tilde{\beta}_{k,\circ}) + \sum_{v=1}^{V}\log\Gamma(\tilde{\beta}_{k,v}) - (\tilde{\beta}_{k,v}-1)\Big(\Psi(\tilde{\beta}_{k,v}) - \Psi(\tilde{\beta}_{k,\circ})\Big)\right)\\
\mathbf{H}(Q(Z)) &= \sum_{m=1}^{M}\sum_{n=1}^{N_m}\sum_{k=1}^{K} -\xi_{m,n}^{k}\log\xi_{m,n}^{k}\ .
\end{aligned}$$

Optimising the ELBO function F given in Equation 2.26 with respect to variational parameters $\tilde{\alpha}$, $\tilde{\beta}$ and ξ gives the following equations:

$$\begin{aligned}
\tilde{\alpha}_{m,k} &= \alpha_k + \sum_{n=1}^{N_m}\xi_{m,n}^{k}\\
\tilde{\beta}_{k,v} &= \beta_v + \sum_{m=1}^{M}\sum_{n=1}^{N_m}\mathbb{1}\left(W_{m,n}=v\right)\xi_{m,n}^{k}\\
\xi_{m,n}^{k} &\propto \exp\Big(\Psi(\tilde{\alpha}_{m,k}) - \Psi(\tilde{\alpha}_{m,\circ}) + \Psi(\tilde{\beta}_{k,W_{m,n}}) - \Psi(\tilde{\beta}_{k,\circ})\Big)\ ,
\end{aligned} \tag{2.27}$$

which guaranty to optimize the ELBO function F at each iteration, and eventually, to converge to a local optima [147].

2.3.2 Gibbs Sampling

Gibbs sampling [54] is a Markov-chain Monte Carlo (MCMC) algorithm, which can be seen as a special case of the Metropolis–Hastings algorithm [101]. It can be used to obtain an observation sequence from a high-dimensional multivariate probability distribution. The sequence can be used to approximate a marginal distribution for one or a subset of the model's variables. In addition, it can be used to compute an integral over one of the hidden variables, and eventually compute its expected value.

In order to build a Gibbs sampler for a model with one multidimensional hidden variable x and observed variable D, full conditionals $P(x_i|x_{\neg i}, D)$ need to be calculated, where, $x_{\neg i}$ represents all other dimensions of variable x excluding the i^{th} dimension. Thus, the Gibbs sampling process involves repetition of two steps:

1. Choose a dimension i (order is not important)
2. Sample x_i from distribution $P(x_i|x_{\neg i}, D)$.

It is possible, for some models, to marginalise over one or more variables analytically. Consequently, collapsed Gibbs sampling (CGS) [91], which is a variant of Gibbs sampling, can be used for the remaining variables. This reduces the complexity of the original model and makes inference simpler without losing the model's generality. In the next section, a CGS for LDA model is elaborated.

2.3.2.1 LDA Collapsed Gibbs Sampler

The first step to apply CGS for a model is to check whether marginalising over some model variable is easy. Thanks to the conjugacy between Multinomial and Dirichlet distributions, a collapsed Gibbs sampler [59] can be implemented for LDA, where the θ and φ variables can be analytically integrated out before carrying out the Gibbs sampling process. This allows us to sample directly from the distribution $P(Z|W, \alpha, \beta)$ instead of the distribution $P(Z, \theta, \varphi|W, \alpha, \beta)$; which reduces the number of hidden variables in the LDA model, and makes the inference and learning process faster.

For LDA, it is required to get samples from the posterior distribution $P(Z|W, \alpha, \beta)$, thus full conditional distributions $P(Z_{d,t}|Z_{\neg(d,t)}, W, \alpha, \beta)$ should be defined, where,

$Z_{\neg(d,t)}$ represents topic assignments for all corpus words after excluding the t^{th} word in document W_d. Assuming that the t^{th} word in document W_d is a word instance of term v, $W_{(d,t)} = v$ then:

$$\begin{aligned} P(Z_{d,t} = k | Z_{\neg(d,t)}, W, \alpha, \beta) &= \frac{P(Z_{d,t} = k, Z_{\neg(d,t)}, W | \alpha, \beta)}{P(Z_{\neg(d,t)}, W_{\neg(d,t)} | \alpha, \beta) P(W_{d,t} | \alpha, \beta)} \\ &\propto \frac{P(Z_{d,t} = k, Z_{d,\neg t}, W_d | \alpha, \beta)}{P(Z_{d,\neg t}, W_{d,\neg t} | \alpha, \beta)} \\ &\propto (\widehat{z_{d,\circ}^{k,\neg(d,t)}} + \alpha_k) \frac{\widehat{z_{\circ,v}^{k,\neg(d,t)}} + \beta_v}{\sum_{r=1}^{V} \widehat{z_{\circ,r}^{k,\neg(d,t)}} + \beta_r} \quad . \end{aligned} \tag{2.28}$$

where, $W_{d,\neg t}$ is words of document W_d excluding its t^{th} word and $Z_{d,\neg t}$ is topic assignments of words $W_{d,\neg t}$. Also, $\widehat{z_{d,\circ}^{k,\neg(d,t)}}$ is the number of words in document W_d assigned to topic k after excluding the document's t^{th} word, whereas $\widehat{z_{\circ,v}^{k,\neg(d,t)}}$ is the number of word instances of term v assigned to topic k from all corpus documents after excluding the t^{th} word in document W_d. Finally, the values of θ and φ, which correspond to a topic setting Z, need to be calculated. By definition, those two variables are distributed Multinomially with Dirichlet priors. Thus, they are distributed by the Dirichlet-Multinomial distribution as follows:

$$P(\theta_d | Z_d, \alpha) \sim Dir(\widehat{z_{d,\circ}} + \alpha)$$
$$P(\varphi_k | Z, \beta) \sim Dir(\widehat{z_{\circ}^{k}} + \beta)$$

where $\widehat{z_{d,\circ}}$ is a vector of topics observation counts in the document W_d and $\widehat{z_{\circ}^{k}}$ is a vector of term observation counts for topic k. Therefore, and using the expectation of the Dirichlet distribution, θ and φ corresponding to the setting Z are given by:

$$\theta_d^k = \frac{\widehat{z_{d,\circ}^{k}} + \alpha_k}{\sum_{i=1}^{K} \widehat{z_{d,\circ}^{i}} + \alpha_i} \tag{2.29}$$

$$\varphi_k^v = \frac{\widehat{z_{d,\circ}} + \beta_v}{\sum_{r=1}^{V} \widehat{z_{\circ,r}^{k}} + \beta_r} \quad . \tag{2.30}$$

Consequently, LDA's collapsed Gibbs sampling algorithm is given by Algorithm 4, where, $\widehat{z_{\circ,\circ}^{k}} = \sum_{r=1}^{V} \widehat{z_{\circ,r}^{k}}$ and $\beta_\circ = \sum_{r=1}^{V} \beta_r$

Algorithm 4 LDA collapsed Gibbs sampler

Input: W words of the corpus, α and β the model hyperparameters.
Output: Z topic assignments, θ topics mixtures, and φ topics distributions.

```
Randomly initialize Z with integers ∈ [1..K]
repeat
  for d = 1 to M do
    for t = 1 to N_d do
      v ← W_{d,t}; k ← Z_{d,t}
      ẑ^k_{d,∘} ← ẑ^k_{d,∘} − 1; ẑ^k_{∘,v} ← ẑ^k_{∘,v} − 1; ẑ^k_{∘,∘} ← ẑ^k_{∘,∘} − 1;
      k ∼ (ẑ^k_{d,∘} + α_k) (ẑ^k_{∘,v} + β_v) / (ẑ^k_{∘,∘} + β_∘)
      Z_{d,t} ← k
      ẑ^k_{d,∘} ← ẑ^k_{d,∘} + 1; ẑ^k_{∘,v} ← ẑ^k_{∘,v} + 1; ẑ^k_{∘,∘} ← ẑ^k_{∘,∘} + 1;
    end for
  end for
until convergence
Calculate θ using Equation 2.29
Calculate φ using Equation 2.30
return Z,θ,φ
```

2.3.2.2 SLDA Collapsed Gibbs Sampler

SLDA is a supervised extension to LDA which uses an added response variable for each document in the corpus in addition to the same component distributions. Thus, a collapsed Gibbs sampler can be implemented for SLDA because it exhibits the same conjugacy between Multinomial and Dirichlet distributions. Starting from the marginal distribution $P(W, Y, Z|\alpha, \beta, \mu, \delta)$ displayed in Equation 2.18, full conditionals for latent variable Z given observed variables and model parameters need to be calculated. In other words, the conditional distributions $P(Z_{(d,t)} = k|Z_{\neg(d,t)}, W, Y, \alpha, \beta, \mu, \delta)$ for each word $W_{d,t}$ need to be defined. Hence:

$$\begin{aligned} P(Z_{d,t}|Z_{\neg(d,t)}, W, Y, \mathcal{H}) &= \frac{P(Z_{d,t}, Z_{\neg(d,t)}, W, Y|\mathcal{H})}{P(Z_{\neg(d,t)}, W_{\neg(d,t)}, Y|\mathcal{H})P(W_{d,t}|\mathcal{H})} \\ &\propto \frac{P(Z_{d,t}, Z_{d,\neg t}, W_d, Y_d|\mathcal{H})}{P(Z_{d,\neg t}, W_{d,\neg t}, Y_d|\mathcal{H})} \quad ; \end{aligned} \tag{2.31}$$

where, $\mathcal{H}$ represents the model's parameters: α, β, μ and δ. Consequently, for a word $W_{d,t}$ and its specific topic assignment $Z_{d,t} = k$, a proportional probability is

given by the following formula:

$$P(Z_{d,t}|Z_{\neg(d,t)}, W, Y, \mathcal{H}) \propto (\widehat{z_{d,\circ}^{k,\neg(d,t)}} + \alpha_k) \cdot \frac{\widehat{z_{\circ,v}^{k,\neg(d,t)}} + \beta_v}{\sum_{r=1}^{V} \widehat{z_{\circ,r}^{k,\neg(d,t)}} + \beta_r} \cdot \exp\left(\frac{\mu_k}{\delta N_d}\left(Y_d - \overline{Z_{d,\neg t}} \cdot \mu - \frac{\mu_k}{2N_d}\right)\right) \quad . \quad (2.32)$$

where, $\overline{Z_{d,\neg t}}$ is the document's W_d updated discrete distribution over topics after excluding the t^{th} word. Starting from a setting for a topic assignments variable Z, both θ and φ variables can be estimated using Equation 2.29 and Equation 2.30 respectively.

Parameters estimation. GLM parameters need to be optimized as part of the inference process. In this thesis, GLM with Gaussian distribution is used; thus, given a setting of variable Z, the corpus level log likelihood for SLDA model parameters μ and δ is given by:

$$\log \mathcal{L}(\alpha, \beta, \mu, \delta|W, Y, Z) = \sum_{k=1}^{K} \log \frac{B(\widehat{z_{\circ}^{k}} + \beta)}{B(\beta)} + \sum_{m=1}^{M} \log \frac{B(\widehat{z_{m,\circ}} + \alpha)}{B(\alpha)} + \log P(Y_m|\overline{Z_m}, \mu, \delta) \quad ; \quad (2.33)$$

where, $P(Y_m|\overline{Z_m}, \mu, \delta)$ is given by Equation 2.15 with normal distribution. Applying the first-order condition on this equation for μ; the partial derivative needed is:

$$\begin{aligned} \frac{\partial \log \mathcal{L}}{\partial \mu} &= \sum_{m=1}^{M} \frac{\partial \log P(Y_m|\overline{Z_m}, \mu, \delta)}{\partial \mu} \\ &= \frac{1}{2\delta} \frac{\partial\Big[(Y - \overline{Z} \cdot \mu)^T (Y - \overline{Z} \cdot \mu)\Big]}{\partial \mu} \\ &= \frac{1}{\delta}\Big(\overline{Z}^T \overline{Z} \cdot \mu - \overline{Z}^T Y\Big) \quad . \end{aligned} \quad (2.34)$$

Where, $\overline{Z}$ is an $M \times K$ matrix with each row $\overline{Z_d}$ representing the document W_d discrete topics' distribution; also Y is an $M \times 1$ vector which contains the observed response values for corpus documents. Consequently, the value which maximizes the

model's log likelihood for a topic setting Z is given by:

$$\mu = \left(\overline{Z}^T \overline{Z}\right)^{-1} \overline{Z}^T Y \quad . \tag{2.35}$$

In addition, a first-order condition should be applied on Equation 2.33 for δ in order to optimize this parameter; this yields after substituting the value of μ from Equation 2.35:

$$\begin{aligned} \delta &= \frac{1}{M}\left(Y - \overline{Z}\mu\right)^T \left(Y - \overline{Z}\mu\right) \\ &= \frac{1}{M}\left(Y^T Y - Y^T \overline{Z}(\overline{Z}^T \overline{Z})^{-1} \overline{Z}^T Y\right) \quad . \end{aligned} \tag{2.36}$$

Full SLDA collapsed Gibbs sampler steps are presented in Algorithm 5; however, α and β parameters are not optimized or sampled in this algorithm. More details about estimating α and β parameters values is presented in section 2.3.4. This works for both LDA and SLDA because of the similarities which those two models share.

Algorithm 5 SLDA collapsed Gibbs sampler

Input: W words of the corpus, Y documents' response values, α, β, μ and δ the model parameters.
Output: Z topic assignments, θ topics mixtures, φ topics distributions.

```
Randomly initialize Z with integers ∈ [1..K]
repeat
  for d = 1 to M do
    for t = 1 to N_d do
      v ← W_{d,t}; k ← Z_{d,t}
      ẑ^k_{d,∘} ← ẑ^k_{d,∘} − 1; ẑ^k_{∘,v} ← ẑ^k_{∘,v} − 1; ẑ^k_{∘,∘} ← ẑ^k_{∘,∘} − 1;
      k ∼ (ẑ^k_{d,∘} + α_k) (ẑ^k_{∘,v} + β_v) / (ẑ^k_{∘,∘} + β_∘) exp((μ_k/(δ N_d))(Y_d − Z̄_{d,¬t} · μ − μ_k/(2 N_d)))
      Z_{d,t} ← k
      ẑ^k_{d,∘} ← ẑ^k_{d,∘} + 1; ẑ^k_{∘,v} ← ẑ^k_{∘,v} + 1; ẑ^k_{∘,∘} ← ẑ^k_{∘,∘} + 1;
    end for
  end for
  μ ← (Z̄^T Z̄)^{-1} Z̄^T Y
  δ ← (1/M)(Y − Z̄μ)^T (Y − Z̄μ)
until convergence
Calculate θ using Equation 2.29
Calculate φ using Equation 2.30
return Z,θ,φ
```

Prediction Given an estimated SLDA model, one can use it to predict a response value for an unseen document. Starting from an unseen document $\tilde{W}_d$, the main objective of SLDA is to predict its response value $\tilde{Y}_d$. Thus, the first step is to apply the inference process on $\tilde{W}_d$ given an already trained SLDA model. This is a traditional LDA inference task because response variable $\tilde{Y}_d$ is unknown for document $\tilde{W}_d$. Once converged, its topic distribution $\overline{\tilde{Z}_d}$ is calculated. Eventually, the response variable of document $\tilde{W}_d$ can be calculated using the following formula:

$$\tilde{Y}_d = \overline{\tilde{Z}_d} \cdot \mu \ . \tag{2.37}$$

Although SLDA supports multi-class classification it does not support multi-label classification, where one document may be assigned to more than one class. Other topic models, such as labelled latent Dirichlet allocation (LLDA), can be used for multi-label classification tasks [127]. In this thesis, SLDA is used as a backup evaluation technique and to investigate how different models perform under supervised tasks such as classification.

2.3.3 Discussion

Variational Bayes and Gibbs sampling are two different approaches to estimate topic models. On the one hand, variational Bayes uses a tractable simple surrogate model which is as close as possible to the true intractable model. Thus, the inference process, in this case, is fast and deterministic; however, it is not optimizing the true model directly and can lose some important dependencies in the process. On the other hand, Gibbs sampling uses MCMC techniques to give samples from the model's true posterior. This preserves important dependencies of the original model; however, in practice, only limited resources are available and only a finite number of samples can be averaged to approximate the intractable model of interest. Moreover, there is no simple way to tell if a number of samples is enough to get a good estimation [136].

Gibbs sampling—despite being slower to converge—is widely considered to provide the most accurate results [147, 110, 137]. However, Asuncion et al. in [9] show

that with appropriate values of hyperparameters α and β, both methods provide the same level of accuracy. Although Gibbs sampling offers samples from the true posterior, a problem arises in practice with averaging multiple samples because there is no guarantee that topic labels are unified. One approach to deal with this is to use an assignment algorithm such as the Hungarian algorithm [83, 84] to match topics of different runs or samples. The sampler is run for at least ten times using different splits to avoid dealing with this problem, as it is essential to use multiple samples from MCMC [118]. The collapsed Gibbs sampling is used for all experiments in this thesis; moreover, all parameters are optimized in order to get the best performance. In the next section, popular estimation techniques for the LDA hyperparameters α and β are explored.

2.3.4 Hyperparameters Estimation

Consider a set of data-count vectors $\mathcal{D} = \{\pi_1, \pi_2, ..\pi_N\}$ where π_j^i is the number of times the outcome was i in the j^{th} sample. Assuming that these data are distributed according to a multivariate Pólya distribution with parameter α, then the best value of this parameter based on observed data is the value which maximizes the following likelihood:

$$\begin{aligned}\mathcal{L}(\alpha|\mathcal{D}) = \prod_{j=1}^{N} P(\pi_j|\alpha) &= \prod_{j=1}^{N} \int_{\rho_j} P(\pi_j|\rho_j) P(\rho_j|\alpha)\, d\rho_j \\ &= \prod_{j=1}^{N} \frac{\Gamma(\alpha_\circ)}{\Gamma(\pi_j^\circ + \alpha_\circ)} \prod_{i=1}^{K} \frac{\Gamma(\pi_j^i + \alpha_i)}{\Gamma(\alpha_i)} \quad .\end{aligned} \tag{2.38}$$

where, $\alpha_\circ = \sum_{i=1}^{K} \alpha_i$ and $\pi_j^\circ = \sum_{i=1}^{K} \pi_j^i$.

The research literature is replete with methods to estimate multivariate Pólya parameters; however, there is no exact closed-form solution available [130, 157]. One of the most accurate methods is Minka's fixed-point iteration approach [106]. However, one of the fastest techniques is the Moments method [106, 130, 88].

2.3.4.1 Ronning's Moments Method

The Moments method, which is an approximate maximum likelihood technique, is particularly useful as an initialization step for other methods. It provides a fast way to learn approximations to Dirichlet or multivariate Pólya distribution parameters directly from data. The Moments method uses known formulae for the target distribution's first and second moments to calculate its parameters. The following formula gives the first moment (mean) of the multivariate Pólya density function:

$$E[\pi^i] = \pi^\circ \frac{\alpha_i}{\alpha_\circ} \quad . \tag{2.39}$$

It is easy to calculate the empirical mean value from data counts; thus, in order to figure out the parameter α, all that is required to calculate is the value of $\alpha_\circ$. This can be done using the second moment (variance) value. The variance of one dimension is enough to calculate $\alpha_\circ$ [44]:

$$var[\pi^i] = \frac{E[\pi^i](\pi^\circ - E[\pi^i])(\pi^\circ + \alpha_\circ)}{\pi^\circ(1 + \alpha_\circ)} \quad , \tag{2.40}$$

gives:

$$\alpha_\circ = \frac{\pi^\circ \left(E[\pi^i]\left(\pi^\circ - E[\pi^i]\right) - var[\pi^i]\right)}{\pi^\circ \left(var[\pi^i] - E[\pi^i]\right) + E[\pi^i]^2} \quad . \tag{2.41}$$

However, In [130] Ronning suggests that using the first $K-1$ dimensions gives more accurate results:

$$\log \alpha_\circ = \frac{1}{K-1} \sum_{i=1}^{K-1} \log \left(\frac{\pi^\circ \left(E[\pi^i]\left(\pi^\circ - E[\pi^i]\right) - var[\pi^i]\right)}{\pi^\circ \left(var[\pi^i] - E[\pi^i]\right) + E[\pi^i]^2} \right) \quad . \tag{2.42}$$

2.3.4.2 Minka's Fixed-point Iteration Method

The idea behind Minka's fixed-point iteration method for maximizing the likelihood is as follows: starting from an initial estimate of the multivariate Pólya distribution parameter α, a simple lower bound on the likelihood, which is tight on α, is constructed. The maximum value of this new lower bound is calculated in a closed form and becomes a new estimate of α [106]. This process is repeated until convergence.

Thus, the objective is to maximize the likelihood function for the multivariate Pólya distribution:

$$\mathcal{L}(\alpha|\mathcal{D}) = \prod_{j=1}^{N} \frac{\Gamma(\alpha_\circ)}{\Gamma(\pi_j^\circ + \alpha_\circ)} \prod_{i=1}^{K} \frac{\Gamma(\pi_j^i + \alpha_i)}{\Gamma(\alpha_i)} \quad . \tag{2.43}$$

The following two lower bounds can be used to facilitate the calculation of the maximum likelihood:

$$\frac{\Gamma(\zeta)}{\Gamma(m+\zeta)} \geq \frac{\Gamma(\hat{x})}{\Gamma(m+\hat{\zeta})} e^{(\hat{\zeta}-\zeta)(\Psi(m+\hat{\zeta})-\Psi(\hat{\zeta}))} \tag{2.44}$$

and,

$$\frac{\Gamma(m+\zeta)}{\Gamma(\zeta)} \geq \frac{\Gamma(\hat{\zeta}+m)}{\Gamma(\hat{\zeta})} \left(\frac{\zeta}{\hat{\zeta}}\right)^{\hat{\zeta}\left[\Psi(\hat{\zeta}+m)-\Psi(\hat{\zeta})\right]} \quad . \tag{2.45}$$

where $m \in \mathbb{Z}_{\geq 0}$ is a positive integer, $\hat{\zeta} \in \mathbb{R}_{>0}$ and $\zeta \in \mathbb{R}_{>0}$ are strictly positive real numbers. The Ψ function is the first derivative of the loggamma function, known as the digamma function [36]:

$$\Psi(x) = \frac{\partial[\log \Gamma(x)]}{\partial x} = \frac{\Gamma'(x)}{\Gamma(x)}$$

Substituting Equation 2.44 and Equation 2.45 in Equation 2.43 leads to:

$$\mathcal{L}(\alpha|\mathcal{D}) \geq \prod_{j=1}^{N} e^{-\alpha_\circ\left[\Psi(\pi_j^\circ+\alpha_\circ^\star)-\Psi(\alpha_\circ^\star)\right]} \prod_{i=1}^{K} {\alpha_i^\star}^{\alpha_i^\star\left[\Psi(\pi_j^i+\alpha_i^\star)-\Psi(\alpha_i^\star)\right]} \cdot C \quad . \tag{2.46}$$

where $\alpha_i^\star$, $\alpha_\circ^\star$ are two real values close to the original values of α_i and $\alpha_\circ$ respectively. The values used here are the previous estimate of α_i and $\alpha_\circ$. And, C is a constant which comprises all terms that do not involve α. Thus, taking the logarithm of both sides of Equation 2.46 leads to:

$$\log \mathcal{L}(\alpha|\mathcal{D}) \geq \mathcal{F}(\alpha) + C \quad , \tag{2.47}$$

where, $\mathcal{F}$ is a function given by the following formula:

$$\mathcal{F}(\alpha) = \sum_{j=1}^{N} -\alpha_\circ \left[\Psi(\pi_j^\circ + \alpha_\circ^\star) - \Psi(\alpha_\circ^\star)\right] + \sum_{i=1}^{K} \log \alpha_i \left[\Psi(\pi_j^i + \alpha_i^\star) - \Psi(\alpha_i^\star)\right] \alpha_i^\star \quad .$$

Consequently, it is possible to find the maximum of this bound in a closed form. Firstly by calculating the derivative of $\mathcal{F}$ with respect to α_i and then solving the equation $\frac{\partial[\mathcal{F}(\alpha)]}{\partial\alpha_i} = 0$:

$$\begin{aligned}\frac{\partial[\mathcal{F}(\alpha)]}{\partial\alpha_i} &= \sum_{j=1}^{N} \frac{\left[\Psi(\pi_j^i + \alpha_i^\star) - \Psi(\alpha_i^\star)\right]\alpha_i^\star}{\alpha_i} - \left[\Psi(\pi_j^\circ + \alpha_\circ^\star) - \Psi(\alpha_\circ^\star)\right] \\ &= 0 \quad .\end{aligned} \tag{2.48}$$

The previous first degree equation has a simple solution:

$$\alpha_i = \alpha_i^\star \frac{\sum_{j=1}^{N} \Psi(\pi_j^i + \alpha_i^\star) - \Psi(\alpha_i^\star)}{\sum_{j=1}^{N} \Psi(\pi_j^\circ + \alpha_\circ^\star) - \Psi(\alpha_\circ^\star)} \quad . \tag{2.49}$$

In [150], Wallach provides a faster version of this algorithm by using the digamma function recurrence relation. This is done by representing data counts samples as histograms. In other words, let N be the number of samples for a K dimensional multivariate Pólya distribution. Then, a more efficient representation would be as K vectors of counts of elements, where the m^{th} cell of the i^{th} vector represents the number of times the count m is observed in the set of N values related to the dimension i. This value is represented by:

$$\mathcal{C}_i^m = \sum_{j=1}^{N} \delta(\pi_j^i - m) \quad ; \tag{2.50}$$

where δ is the Dirac function. Similarly, $\mathcal{C}_\circ^m$ represents the number of times the sum m is observed in the set of N sum values—over all dimensions—of sample counts.

$$\mathcal{C}_\circ^m = \sum_{j=1}^{N} \delta\left(\pi_j^\circ - m\right) \quad , \tag{2.51}$$

where, $\pi_j^\circ = \sum_{i=1}^{K} \pi_j^i$. Equation 2.50 and Equation 2.51 allow us to rewrite Equation 2.49 in a more efficient way:

$$\alpha_i = \alpha_i^\star \frac{\sum_{m=1}^{dim(\mathcal{C}_i)} \mathcal{C}_i^m \left[\Psi(m + \alpha_i^\star) - \Psi(\alpha_i^\star)\right]}{\sum_{m=1}^{dim(\mathcal{C}_\circ)} \mathcal{C}_\circ^m \left[\Psi(m + \alpha_\circ^\star) - \Psi(\alpha_\circ^\star)\right]} \quad , \tag{2.52}$$

where, $\mathcal{C}_i$ is a histogram vector for all counts in the N samples associated to the dimension i and $\mathcal{C}_\circ$ is a histogram vector for all count sums over all dimensions in the N samples. And, $dim(\mathcal{C}_i)$ and $dim(\mathcal{C}_\circ)$ are the numbers of elements in vectors $\mathcal{C}_i$ and $\mathcal{C}_\circ$ respectively. This new formula speeds the computation to an extent that depends on how many frequent count values can be spotted in each dimension $i \in [1..K]$. The more frequent values there are, the faster the computation is. Unfortunately, the digamma function call is time-consuming in practice; however, in [150], Wallach suggests that there is room for improving the performance by getting rid of the digamma function call completely. This can be done by taking into consideration the digamma recurrence relation in [36]:

$$\Psi(x+1) = \Psi(x) + \frac{1}{x} \quad . \tag{2.53}$$

This formula can be extended for any positive integer m:

$$\Psi(x+m) = \Psi(x) + \sum_{l=1}^{m} \frac{1}{x+l-1} \quad . \tag{2.54}$$

Rewriting gives:

$$\Psi(x+m) - \Psi(x) = \sum_{l=1}^{m} \frac{1}{x+l-1} \quad . \tag{2.55}$$

Substituting Equation 2.55 in Equation 2.52 leads to:

$$\alpha_i = \alpha_i^\star \frac{\sum_{m=1}^{dim(\mathcal{C}_i)} \mathcal{C}_i^m \sum_{l=1}^{m} \frac{1}{\alpha_i^\star + l - 1}}{\sum_{m=1}^{dim(\mathcal{C}_\circ)} \mathcal{C}_\circ^m \sum_{l=1}^{m} \frac{1}{\alpha_\circ^\star + l - 1}} \quad . \tag{2.56}$$

An efficient implementation of Minka's fixed-point iteration using Equation 2.56 is listed in Algorithm 6.

2.4 Evaluation of Topic Models

Due to the unsupervised nature of LDA-based topic modelling algorithms, the evaluation of inferred topic models is a difficult task. However, there are some popular methods in the literature to attempt this evaluation. A topic model's 'perplexity',

Algorithm 6 Minka fixed-point iteration method

Input: $\mathcal{C}$ samples counts histograms, $\mathcal{C}_\circ$ samples lengths histogram.
Output: α the parameter for multivariate Pólya distribution.

```
Initialize α using Equation 2.39 and Equation 2.42 (the Moments method).
repeat
    Dgma ← 0
    Dntr ← 0
    for m = 1 to dim(C_∘) do
        Dgma ← Dgma + 1/(α_∘ + m − 1)
        Dntr ← Dntr + C_∘^m Dgma
    end for
    for i = 1 to K do
        Dgma ← 0
        Nmtr ← 0
        for m = 1 to dim(C_i) do
            Dgma ← Dgma + 1/(α_i + m − 1)
            Nmtr ← Nmtr + C_i^m Dgma
        end for
        α_i ← α_i Nmtr/Dntr
    end for
until convergence
return α
```

under a hold-out set of test documents, is usually used as a standard evaluation metric. Moreover, a topic model's performance in a supervised task can also be used to benchmark its performance against other models. In addition, learnt topics coherence is often used as a metric to evaluate the sensibility of interred topics. These methods are further described below.

2.4.1 Perplexity

One common way to evaluate a topic model is to calculate its perplexity under a set of unseen test documents. Perplexity is a measure to benchmark a topic model's ability to generalize to unseen documents. In other words, it provides a numerical value indicating, in effect, how much the topic model is 'surprised' by new data. The higher the probability of test document words given the model, the smaller the perplexity value becomes. Consequently, a model with a smaller perplexity value can be considered to have a better ability to generalize to unseen documents.

Let $\tilde{W}$ be an unseen test corpus which contains $\tilde{M}$ documents. The perplexity is calculated by exponentiating the negative mean log-likelihood value of the whole

set of document words given the model. The following formula gives perplexity:

$$Perplexity(\tilde{W}|W, Z, \alpha, \beta) = \exp\left(\frac{-\sum_{j=1}^{\tilde{M}} \log P(\tilde{W}_j|W, Z, \alpha, \beta)}{\sum_{j=1}^{\tilde{M}} \tilde{N}_j}\right) \tag{2.57}$$

where $j \in [1..\tilde{M}]$ and $\tilde{N}_j$ is the number of words in test document $\tilde{W}_j$. Unfortunately the exact value of the marginal distributions $P(\tilde{W}_j|W, Z, \alpha, \beta)$ is intractable due to the need to sum over all different topic assignments settings for test corpus words. However, there are multiple methods to approximate this marginal probability in the literature such as: annealed importance sampling (AIS) [111], harmonic mean method [117], Chib-style estimation [27, 151] and Left-To-Right algorithm [152, 22]. Left-To-Right is one of the best methods in the literature and is described next.

2.4.1.1 The Left-To-Right Algorithm

The Left-To-Right method is based on breaking the problem into a series of parts:

$$\begin{aligned} P(\tilde{W}_j|W, Z, \alpha, \beta) &= \prod_{t=1}^{\tilde{N}_j} P(\tilde{W}_{j,t}|\tilde{W}_{j,1}, \tilde{W}_{j,2}, ..., \tilde{W}_{j,t-1}, W, Z, \alpha, \beta) \\ &= \prod_{t=1}^{\tilde{N}_j} \sum_{\tilde{Z}_{j,1},...,\tilde{Z}_{j,t}} P(\tilde{W}_{j,t}, \tilde{Z}_{j,1}, ..., \tilde{Z}_{j,t}|\tilde{W}_{j,1}, ..., \tilde{W}_{j,t-1}, W, Z, \alpha, \beta) \ . \end{aligned} \tag{2.58}$$

where $\tilde{Z}_j$ gives the topic assignments of test document $\tilde{W}_j$. It can be seen that the previous equation involves marginalizing out the variable $\tilde{Z}_j$; this is intractable for large test documents and a high number of topics K. Luckily, the previous sum over all possible value settings of $\tilde{Z}_{j,1}, ..., \tilde{Z}_{j,t}$ can be approximated using sequential Monte Carlo techniques [39] with R particles. Thus, let $(\tilde{Z}_{j,1}, ..., \tilde{Z}_{j,t}) \sim P\left(\tilde{Z}_{j,1}, ..., \tilde{Z}_{j,t}|\tilde{W}_{j,1}, ..., \tilde{W}_{j,t-1}, W, Z, \alpha, \beta\right)$. Consequently, the approximation can be

calculated using R samples from the previous distribution as follows:

$$\begin{aligned}
&\sum_{\tilde{Z}_{j,1},...,\tilde{Z}_{j,t}} P(\tilde{W}_{j,t}, \tilde{Z}_{j,1}, ..., \tilde{Z}_{j,t} | \tilde{W}_{j,1}, ..., \tilde{W}_{j,t-1}, W, Z, \alpha, \beta) \\
&\qquad \approx \frac{1}{R} \sum_{r=1}^{R} P(\tilde{W}_{j,t} | \tilde{W}_{j,1}, ..., \tilde{W}_{j,t-1}, \tilde{Z}^r_{j,1}, ..., \tilde{Z}^r_{j,t-1}, W, Z, \alpha, \beta) \\
&\qquad = \frac{1}{R} \sum_{r=1}^{R} \sum_{k=1}^{K} \frac{\widehat{z^k_{\circ,\tilde{W}_{j,t}}} + \beta_{\tilde{W}_{j,t}}}{\widehat{z^k_{\circ,\circ}} + \beta_\circ} \frac{\widehat{\tilde{z}^{k,r}_{j,\circ}} + \alpha_k}{\sum_{i=1}^{K} \widehat{\tilde{z}^{i,r}_{j,\circ}} + \alpha_i} \quad ,
\end{aligned} \tag{2.59}$$

where $\widehat{\tilde{z}^{i,r}_{j,\circ}}$ is the number of words in test document $\tilde{W}_j$ that are assigned to topic i in the sample r, and $\tilde{Z}^r_{j,t}$ is the topic assignment for the t^{th} word in test document $\tilde{W}_j$ and sample r. The Left-To-Right algorithm is given by Algorithm 7.

Algorithm 7 The Left-to-right algorithm to estimate the value $\log P(\tilde{W}_j | W, Z, \alpha, \beta)$

Input: W words of the training corpus, $\tilde{W}_j$ words of the j^{th} test document, Z topic assignments of the training corpus, α and β the model's parameters.
Output: $l = \log P(\tilde{W}_j | W, Z, \alpha, \beta)$ the log likelihood of the test document $\tilde{W}_j$ given a trained LDA model.

$l \leftarrow 0$
for $t = 1$ **to** $\tilde{N}_j$ **do**
  $P_t \leftarrow 0$
  for $r = 1$ **to** R **do**
    for $t' = 1$ **to** t **do**
      $v \leftarrow \tilde{W}_{j,t'}$; $k \leftarrow \tilde{Z}_{j,t'}$
      $\widehat{\tilde{z}^k_{j,\circ}} \leftarrow \widehat{\tilde{z}^k_{j,\circ}} - 1$
      $k \sim (\widehat{\tilde{z}^k_{j,\circ}} + \alpha_k) \frac{\widehat{z^k_{\circ,v}} + \beta_v}{\widehat{z^k_{\circ,\circ}} + \beta_\circ}$
      $\tilde{Z}_{j,t'} \leftarrow k$
      $\widehat{\tilde{z}^k_{j,\circ}} \leftarrow \widehat{\tilde{z}^k_{j,\circ}} + 1$
    end for
    $P_t \leftarrow P_t + \sum_{k=1}^{K} \frac{\widehat{z^k_{\circ,\tilde{W}_{j,t}}} + \beta_{\tilde{W}_{j,t}}}{\widehat{z^k_{\circ,\circ}} + \beta_\circ} \frac{\widehat{\tilde{z}^k_{j,\circ}} + \alpha_k}{\sum_{i=1}^{K} \widehat{\tilde{z}^i_{j,\circ}} + \alpha_i}$
  end for
  $l \leftarrow l + \log \frac{P_t}{R}$
  $k \sim (\widehat{\tilde{z}^k_{j,\circ}} + \alpha_k) \frac{\widehat{z^k_{\circ,\tilde{W}_{j,t}}} + \beta_{\tilde{W}_{j,t}}}{\widehat{z^k_{\circ,\circ}} + \beta_\circ}$
  $\widehat{\tilde{z}^k_{j,\circ}} \leftarrow \widehat{\tilde{z}^k_{j,\circ}} + 1$
  $\tilde{Z}_{j,t} \leftarrow k$
end for
return l

2.4.2 Coherence

Unfortunately, perplexity does not always correlate with human judgement about topic quality [104, 25]. Consequently, other tests, such as word-intrusion and topic-intrusion, are introduced in order to evaluate the semantic coherence of the inferred topics[25]. However, Newman et al. provide in [115] an automatic metric to evaluate topics which reflects topics semantic coherence. This evaluation metric is described next.

2.4.2.1 Pointwise Mutual Information

Pointwise Mutual Information (PMI) [30] is an ideal measure of semantic coherence, based on word association in the context of information theory [145, 142]. PMI compares the probability of seeing two words together with the probability of observing the words independently. PMI for two words can be given using the following formula:

$$Pmi(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}. \tag{2.60}$$

The joint probability $P(w_i, w_j)$ can be measured by counting the number of observations of words w_i and w_j together in the corpus normalized by the corpus size. PMI-based evaluations correlate very well with a human judgement of topic coherence or topic semantics [116, 115], especially when Wikipedia is used as a meta-documents to calculate the word co-occurrences within a suitably sized sliding window.

PMI values fall in the range $]-\infty, -\log P(w_i, w_j)]$, hence the higher the PMI value the more coherent the topic it represents. PMI values can be normalized to fall in the range $[-1, 1]$ as shown in [20] using the following formula:

$$nPmi(w_i, w_j) = \begin{cases} -1 & \text{if } P(w_i, w_j) = 0 \\ \frac{\log P(w_i) + \log P(w_j)}{\log P(w_i, w_j)} - 1 & \text{otherwise} \quad . \end{cases} \tag{2.61}$$

The approach used to evaluate one topic is to calculate the mean of PMI for each possible word pair in the top words set of topic φ_k. Consequently, the normalized

PMI value for one topic φ_k is given using the following formula:

$$Coherence(T^k) = \frac{\sum_{w_i,w_j \in T^k} nPmi(w_i, w_j)}{\binom{T^k_{len}}{2}} \quad . \tag{2.62}$$

where, T^k is a set of top words of topic φ_k and T^k_{len} represents the number of words inside words set T^k.

2.4.3 Supervised task Performance

Another way to evaluate a topic model is to use it in a supervised information retrieval task such as classification or spam filtering [156]; then based on its performance accuracy, a topic model can be benchmarked against other models. There are multiple ways to use a topic model in classification. For example, a topic model can be used as a document dimensionality reduction technique to choose features and then carry out classification using standard methods [18]. In this thesis two approaches are used: the first approach is the SLDA model which is described in section 2.2.2.1; and the second approach is to use the 'Multi-Corpus LDA' [14, 15]; the latter approach is described next on a spam filtering task.

2.4.3.1 Multi-Corpus LDA

In the Multi-corpus LDA (MC-LDA) approach [14, 15], two distinct LDA models are inferred using the same vocabulary words. The first model is inferred from the collection of spam documents with $K^{(s)}$ topics, whereas the second model is inferred from the collection of non-spam documents with $K^{(n)}$ topics. Consequently, the word distributions for $K^{(n)} + K^{(s)}$ topics are learned. The idea behind MC-LDA is to merge the previous two models and create a unified model with $K^{(n)} + K^{(s)}$ topics. This is done by simply encoding the topic identification numbers of the spam topic model to begin from $K^{(n)}+1$ instead of beginning from 1. Thus, for an unseen document $\tilde{W}_{\tilde{d}}$, the inference in the unified model can be made using the following

formula:

$$P(\tilde{Z}_{(\tilde{d},t)} = k|\tilde{Z}_{\neg(\tilde{d},t)}, \tilde{W}, W, Z, \alpha, \beta) \propto (\widehat{\tilde{z}_{\tilde{d},\circ}^{k,\neg(\tilde{d},t)}} + \alpha_k)\frac{\widehat{z_{\circ,v}^{k,\neg(\tilde{d},t)}} + \beta_v}{\sum_{r=1}^{V}\widehat{z_{\circ,r}^{k,\neg(\tilde{d},t)}} + \beta_r} \quad , \tag{2.63}$$

where $\widehat{\tilde{z}_{\tilde{d},\circ}^{k,\neg(\tilde{d},t)}}$ represents the number of words in test document $\tilde{W}_{\tilde{d}}$ that are assigned to topic k excluding the t^{th} word in that document. However, the count $\widehat{z_{\circ,v}^{k,\neg(\tilde{d},t)}}$, which represents the number of word instances of vocabulary term v form all documents assigned to topic k, is unknown. Thus the previous Multi-Corpus inference formula's second factor can be approximated using the φ_k^v value. Let $\tilde{W}_{(\tilde{d},t)}$, which is the t^{th} word in test document $\tilde{W}_{\tilde{d}}$, be v, then:

$$P(\tilde{Z}_{(\tilde{d},t)} = k|\tilde{Z}_{\neg(\tilde{d},t)}, \tilde{W}, W, Z, \alpha, \beta) \propto\!\!\!\propto (\widehat{\tilde{z}_{\tilde{d},\circ}^{k,\neg(\tilde{d},t)}} + \alpha_k)\varphi_k^v \quad . \tag{2.64}$$

As a result of the inference process and after a sufficient number of iterations, the words topic assignment $Z_{\tilde{d}}$ is calculated. Consequently, the document topic distribution $\theta_{\tilde{d}}$ is calculated using:

$$\theta_{\tilde{d}}^k = \frac{\widehat{\tilde{z}_{\tilde{d},\circ}^{k}} + \alpha_k}{\sum_{i=1}^{K}\widehat{\tilde{z}_{\tilde{d},\circ}^{i}} + \alpha_i} \quad . \tag{2.65}$$

In order to classify a document $\tilde{W}_{\tilde{d}}$, the LDA prediction value $\tau = \sum_{i=K^{(n)}+1}^{K^{(n)}+K^{(s)}} \theta_{\tilde{d}}^i$ is calculated. if the LDA prediction value τ is above than a specific threshold, the document will be classified as spam. Otherwise, the document can be classified as legitimate.

2.4.4 Other Implementations

Gibbs sampling is used for all topic models provided in this thesis. Although Gibbs sampling is an efficient technique to sample from the true posterior [147], providing other implementation techniques might be useful. Other techniques include variational Bayes [18, 9] and spectral methods which can provide a faster means to estimate topic models. Spectral methods in topic modelling are getting more popu-

lar in the literature [154, 82, 75, 7, 6, 8, 43]. Most of these techniques can learn topic models faster compared with pure Bayesian techniques. For example, the spectral implementation of SLDA proposed in [154] performs faster than other implementations using the Gibbs sampling technique and provides a higher accuracy when it is used to initialize an SLDA Gibbs sampler. Thus it is tempting to investigate the use of spectral dimensionality reduction techniques to implement the methods provided in this thesis.

2.5 Multi-Objective Optimization

In many real-life engineering problems, there is more than one objective to minimize or maximize. These objectives usually conflict with each other; hence, optimizing one objective only may lead to impractical solutions [97]. This kind of problem is called a 'multi-objective optimization problem', which contains multiple objective functions and a set of constraints [103, 73]. Consequently, a multi-objective problem (MOP) can be defined as a function $\mathcal{F} : \Omega \rightarrow S$ as follows [103]:

$$\text{minimize } \mathcal{F}(x) = (f_1(x), .., f_m(x))^T \ , \tag{2.66}$$

where Ω is a non-empty decision space and $S \subset \mathbb{R}^m$ is the objective space; with $m \geq 2$. Multi-objective optimization is the process of solving MOPs, it is not as straightforward as a single-objective optimization; moreover, the problem becomes challenging when objectives conflict. Researchers devise multiple approaches to tackle this kind of problem. These approaches mainly fall into four classes: no preference, priori, posteriori, and interactive methods [73]. In 'no preference' methods, a natural compromise between objectives is specified in advance; thus, there is no need for preference information to be provided. However, for the rest of the classes, preference information is needed at some point.

In this thesis, posteriori preference is used; thus, an approximation set of Pareto optimal solutions is calculated and then the decision maker can choose the desired model. Pareto optimal solutions achieve a trade-off between the problem's objec-

tives; hence, any improvements done for one objective results in worsening of at least one other objective. Let u, $s \in S$; u is said to be dominated by s if $s_i \geq u_i$ for every $i \in [1, .., m]$ and $s_j > u_j$ for at least one $j \in [1, .., m]$. Solution $x \in \Omega$ is a Pareto optimal if there is no other solution $y \in \Omega$ such that $\mathcal{F}(y)$ dominates $\mathcal{F}(x)$; in other words, solution x is not dominated by any other solution in the decision space [37]. The set of all Pareto optimal solutions is called a *Pareto set* (PS), and the set of all objective vectors corresponding to PS is called a *Pareto front* (PF) [103].

There are multiple techniques to tackle a multi-objective problem [10]; some of the classic approaches are: weighted sum method [50], the ε-constraint method [62] and Benson's algorithm [13]. Mainly, posteriori preference methods fall into two classes: mathematical programming and evolutionary algorithms. The main advantage of evolutionary algorithms over the mathematical programming approach is that an evolutionary algorithm can provide an approximation of PS with only one run. However, mathematical approaches need to be run multiple times to generate the optimal set. On the downside, evolutionary algorithms usually require more resources than mathematical approaches.

2.5.1 Topic Modelling as a Multi-Objective Problem

Topic modelling is a MOP as it involves optimizing multiple criteria functions. For example, a topic model should be able to generalize to unseen documents and at the same time provides sensible topics for a human being. The ability to generalize to unseen documents is given by the model's held-out perplexity, whereas topics sensibility can be characterised by coherence; held-out perplexity and coherence do not always correlate [104, 25]. Consequently, by learning a topic model as a MOP; one can find a trade-off between these conflict objectives or choose the model with the "right" characteristics for a specific application. Moreover, some studies show that using multi-objective optimization yields better results than single-objective optimization even for single-objective problems (SOPs) [155, 78].

2.5.2 Multi-Objective Evolutionary Algorithms (MOEAs)

Multi-objective evolutionary algorithms (MOEAs) are well-suited techniques to find good solutions for complex MOPs with two or three objectives [148, 31]. They tackle a MOP by simulating the basic principles of the evolution process on a set of initial solutions, which is treated as an evolving population. Eventually—using Pareto dominance as guidance for selection—initial solutions 'evolve' into a good approximation of the PS after applying evolutionary operations such as: selection, fitness assignment, crossover, mutation and elitism [31, 37]. Because of its flexibility, many MOEAs are developed in the literature; each one handles a MOP differently [162, 148].

2.5.2.1 MOEA/D

MOEA/D [161] is a general framework to solve MOPs which employs decomposition in order to find a good approximation for the PS. It splits the problem into many simpler SOPs and then evolves them simultaneously. There are many techniques for transforming a MOP to a SOP including: weighted sum approach [103], Tchebycheff approach [74, 141], and boundary intersection (BI) variants [35, 100]. In this thesis, Tchebycheff approach is used, thus the Tchebycheff approach is described first.

Tchebycheff Approach Tchebycheff is a decomposition approach to transform a MOP to a SOP [74, 141]. It can be used to transform the MOP defined by Equation 2.66 using the following equation:

$$\text{minimize } \mathcal{G}^{te}(x|\lambda, s^*) = \max_{j\in[1..m]} \{\lambda_j |f_j(x) - s_j^*|\} \quad . \tag{2.67}$$

where, s^* is the ideal point such that $s_j^* = \min(f_j(x))$ for all $x \in S$ and λ is a weight vector which can be represented as a point on the simplex Δ^m. Different weight vector values allow different points from PF to be found; hence, a multi-objective algorithm based on Tchebycheff approach should use various weight vectors. The main advantage of this approach over other approaches such as the weighted sum approach is that the Tchebycheff approach can find points from non-convex concave

PF.

Algorithm 8 MOEA/D Framework

Input: MOP, N number of sub-problems, and T neighbourhood size.
Output: EP an approximation of PS.

```
EP ← ∅
Compute N uni-formally spread weight vectors λ = {λ_1, λ_2, .., λ_N}
for i = 1 to N do
    NB_i ← indices of T nearest weight vectors to λ_i
    x_i ← random vector ∈ Ω
    FV_i ← F(x_i)
end for
Initialize s* using a problem-specific method.
repeat
    for i = 1 to N do
        Randomly select two indices k and l from NB_i
        Generate new solution y using genetic operations on x_k and x_l
        Improve and update solution y using a problem-specific methods
        for j = 1 to m do
            if s*_j > f_j(y) then
                s*_j ← f_j(y)
            end if
        end for
        for j ∈ NB_i do
            if G^te(y|λ_j, s*) < G^te(x_j|λ_j, s*) then
                x_j ← y
                FV_j ← F(y)
            end if
        end for
        nonDominated ← True
        for s ∈ EP do
            if s is dominated by y then
                Remove s from EP
            end if
            if y is dominated by s then
                nonDominated ← False
            end if
        end for
        if nonDominated then
            EP ← EP ∪ y
        end if
    end for
until stopping criteria is met
return EP
```

MOEA/D Framework Let $\lambda = (\lambda_1, \lambda_2, .., \lambda_N)$ be an N evenly spread weight vectors over the standard simplex Δ^m, and s^* is the reference point. Consequently,

the original MOP can be transformed to N single objective sub-problems using the Tchebycheff approach. The i^{th} scalar optimization sub-problem is given by Equation 2.67. It is clear that $\mathcal{G}^{te}$ is continuous on λ; hence, if λ_i is close to λj then $\mathcal{G}^{te}(x|\lambda_i, s^*)$ and $\mathcal{G}^{te}(x|\lambda_j, s^*)$ are close to each other. Therefore, each scalar sub-problem can make use of the neighbourhood information to evolve faster [161]. Thus, MOEA/D algorithm calculates the neighbourhood NB_i of a weight vector λ_i which is a list of T nearest weight vectors from λ_i. Then, solutions corresponding to the neighbour weight vectors are exploited to get useful information for solving the i^{th} sub-problem. In each iteration, MOEA/D maintains a list of current solution for all sub-problems as the only population. Moreover, the reference point s^* which represents the ideal solution is updated based on the current population. MOEA/D framework is given by the Algorithm 8.

2.6 Corpora

In this thesis, topic models are evaluated using multiple corpora. Each one has different characteristics in terms of the number of documents, the length of each document, and the variety of topics. Corpora used in this thesis falls in two categories: unlabeled corpora for unsupervised evaluation and labelled corpora for supervised performance benchmark.

2.6.1 Unlabeled Corpora

The following corpora are used for unsupervised evaluation in this thesis:

1. Wiki corpus: which is a small corpus generated from Wikipedia, it comprises mainly four distinct topics (Love, Music, Sport and Government). This corpus is used in this thesis as a proof of concept. Because it is small, many techniques can be applied easily before they are tested with larger corpora.

2. News Corpus: it is made from about 15000 documents taken from news articles covering mainly four topics: Music, Economy, Fuel and Brain Surgery. This

corpus comprises a small number of topics and a relatively high number of documents.

3. EPSRC Corpus: it contains about 800 documents that are summaries of projects in Information and Communication Technology (ICT) funded by the Engineering and Physical Sciences Research Council (EPSRC). Thus, it exhibits a variety of topics and consequently more challenging for topic modelling. Each one of its documents has the average length of 200 words.

4. NewsAP corpus: a subset of news articles from Associated Press (AP) data from the First Text Retrieval Conference (TREC-1) [65]. It has 38,500 unique terms and 453,462 words spread over 2,213 documents with an average document size of 200 words. NewsAP is rich with topics in a diversity of subjects including politics, surgery, fashion, trading and many others.

5. PubMed Corpus: this is the most extensive corpus used in this thesis, it comprises 4,155,256 documents with 229,742,438 words, and 2,421,771 unique terms. This corpus is a subset of articles abstracts published by National Library of Medicine (NLM) [120]. The average document length in this corpus is only 55 words. A subset of this corpus with relatively larger documents is used in Chapter 5. This contains 70,287 documents with 6,570,235 words and 125,652 unique terms. The average length of each document in this subset is about 93 words. As a topic modelling problem, PubMed corpus might be the most challenging one as it has many topics in the same main subject area. Same words would tend to appear in multiple topics as the inferred topic are close to each other on the semantic level. Consequently, it might be harder for a topic model to distinguish topics.

2.6.2 Labeled Corpora

The following corpora are used in classification tasks, which require each document in the corpus to have a label or tag:

1. Reuters corpus: Reuters-21578, Distribution 1.0 (ModApte split 10 categories)

is a collection of stories that appeared on Reuters newswire in 1987. The corpus is manually tagged and categorized by personnel in Reuters Ltd. It comprises 9,980 documents spread over ten categories.

2. Enron corpus [102], which comprises a subset of Enron emails from the period from 1999 until 2002. This corpus contains 16545 legitimate messages and 17169 spam. Enron is useful for binary classification tasks.

3. LingSpam corpus [135], which contains 2412 legitimate messages and 481 spam. The corpus contains 1,970,249 words with average document length of 680 words.

4. The SMS Collection v.1 [5], which contains 4827 legitimate SMS messages and 747 spam SMS messages. This corpus has shorter documents which might introduce a challenge to topic models.

2.7 Conclusions

This chapter provided background on topic modelling including basic supervised and unsupervised topic models, their inference methods, topic models hyperparameters estimation and topic models evaluation techniques. In addition, background on Multi-objective optimization, which covered the basic principles of multi-objective optimization and MOEAs, was addressed. It showed that topic modelling can be considered as a MOP which has at least two objectives: the ability to generalize to unseen documents, and topics coherence. The next chapter describes the design of an MOEA to solve topic modelling problems by treating them as MOPs.

Chapter 3

Multi-Objective Topic Models

In this chapter, a new topic modelling approach based on Multi-objective evolutionary algorithms (MOEAs), is developed [76]. There are two settings for this new model: the first setting is entirely based on MOEA and starts from scratch; whereas, in the second setting, the optimization starts with an estimated LDA model. To evaluate this model, topic coherence is calculated for the resulting topics, and the model's ability to generalize to unseen documents is measured. The new model exhibits an enhancement in terms of topic coherence. However, no improvement is witnessed in terms of the ability to generalize to unseen documents. In addition, this chapter provides a novel genetic algorithm (GA) to maximize the LDA model's log-likelihood directly by changing words' topic assignments. In spite of being able to optimize the LDA model's log-likelihood, the perplexity score is slightly deteriorated as the number of topics grows.

3.1 Introduction

Current topic modelling approaches such as Latent Dirichlet Allocation (LDA) [18] and Correlated Topic Models (CTM) [16], rely on finding a set of topics that maximizes the likelihood that the data were generated by a specific model of document generation. Though commonly returning interpretable results, the inferred models are ultimately aligned to a much-simplified abstraction of the real document generation process, and leave much room for improvement in the intuitive 'real-world

coherence' of the resulting models. A high quality topic model is one that can be expected to score well on a collection of different criteria, concerned with, for example, the coherence of individual topics, the coherence of the collection of topics as a whole, and the extent to which the inferred topics cover the entire collection, as well as the extent to which individual documents are explained by the topics (for example, a poor topic model in the latter respect may leave large portions of many documents unallocated to topics). However, each of these objectives is difficult to evaluate and can only be approximated; meanwhile, the familiar LDA perplexity criterion is a proven successful objective that, similarly, provides an appropriate and alternative approximate measure of quality.

Exploiting the multi-criteria nature of topic models, in this chapter, firstly the use of multi-objective evolutionary algorithms (MOEAs) in topic modelling is explored, and then it is investigated whether MOEA or MOEA/LDA hybrid approaches can be designed that yield better topic models than current approaches, and consequently provide enhanced effectiveness and user experiences in the many applications of topic modelling technologies.

The remainder of this chapter is organized as follows: in section 3.2, a novel MOEA approach to topic modelling is introduced; later, section 3.3 describes a series of experiments, that compare MOEA-TM approaches with LDA on three text corpora. In section 3.4, a genetic algorithm (GA) to optimize an LDA model's log-likelihood directly is elaborated. Eventually, Summary and final reflections are made in section 3.5. Meanwhile, source code, corpora, and associated instructions that are sufficient to replicate the experiments and support further investigations are provided at http://is.gd/MOEATM.

3.2 MOEA Topic Modelling

Multi-objective optimization aims to find a set of solutions that represent optimal trade-offs between the objectives. This is the set of *Pareto Optimal* solutions [123]. There are a wide variety of approaches to multi-objective problems; however, many of these may fail when the Pareto front (the geometric structure of the Pareto set

in objective space) is concave or disconnected [32]. Multi-objective Evolutionary Algorithms (MOEAs) tend to avoid these drawbacks [32, 33], among others and are currently prominent among state of the art approaches to multi-objective optimization.

Topic models have many applications beyond unstructured text processing and text tagging. They can be used in analyzing genetic data [26], computer vision [95], audio and speech engineering [77, 53], emotion modeling and social affective text mining [11], and financial analysis [45]. Current approaches such as LDA focus on producing topic models which score well on perplexity as measured over a test set. However, other applications, such as text tagging which is used in digital libraries, require highly coherent topics [116]. Considering the varied requirements of other applications, along with arguments made in section 3.1, it is well-worth considering MOEAs in attempt to produce high-quality topic models in general, and also in contexts relating to specific applications.

3.2.1 MOEA Approaches to Topic Modelling

The first approach ('standalone' MOEA-TM) is to optimize two objectives: PMI and coverage (described in section 3.2.4). PMI encourages coherent topics, whilst coverage encourages a large proportion of the corpus words to appear in the inferred topics. In 'standalone' MOEA-TM, the number of words per topic is limited. This arguably leads to more intuitive topics, and significantly reduces computational load, but means that perplexity cannot be used as an objective since the perplexity calculation requires all corpus words to be assigned to a topic. Experiments with standalone MOEA-TM are described in section 3.3.2. An alternative approach is introduced in section 3.3.3 in which MOEA-TM is used to improve topic models pre-generated by LDA. Here the computational load of an unlimited number of words per topic is traded against the optimized starting point, and perplexity is added as an additional objective. In each case, MOEA-TM builds on the current prominent 'Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D)' [161] which is illustrated in section 2.5.2.1, and adapts it to this task.

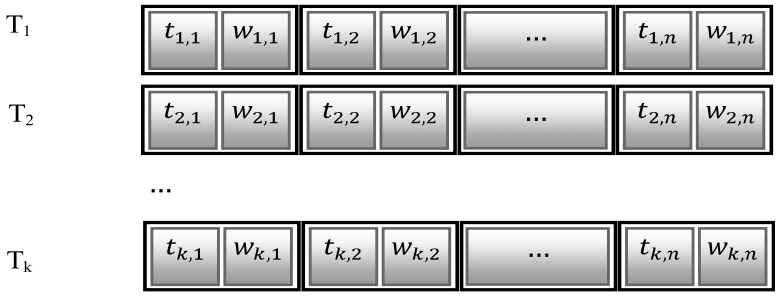


Figure 3.1: Chromosome Structure

3.2.2 Encoding and Generation of Initial Population

Each chromosome is a vector of topic variables $T_1, T_2.., T_K$ where K is the number of topics. Each topic variable is defined as a set of weighted words. Thus, each gene comprises two parts: the word index and a numerical value representing the word's participation in the topic. The Chromosome structure is illustrated in Figure 3.1. In the standalone case, the population is initialized randomly as each topic variable is initialized on the basis of a randomly chosen document. Topic genes are initialized based on the most frequent words in the chosen document, with random weights. However, when the algorithm is used to enhance an existing model, the population echoes the model itself. Each topic variable is based on its corresponding model's topic, where the genes represent the highest weighted words in that topic.

3.2.3 Genetic Operators

Crossover in our approach generates two offspring from two parents. Each child comprises as many topic variables as its parents have, via uniform crossover of the parents' corresponding topic variable genes, ensuring that words and their associated weights are copied together. However, when a word exists in both parents' topic variables, the children have the average word weight. A simple two topics crossover example is illustrated in Figure 3.2.

Mutation is applied to a single randomly chosen gene, changing the weight to a new random number, and changing the word to another word from the corpus, ensuring that the newly introduced word occurs together in a document in the corpus with another randomly selected word from the topic variable.

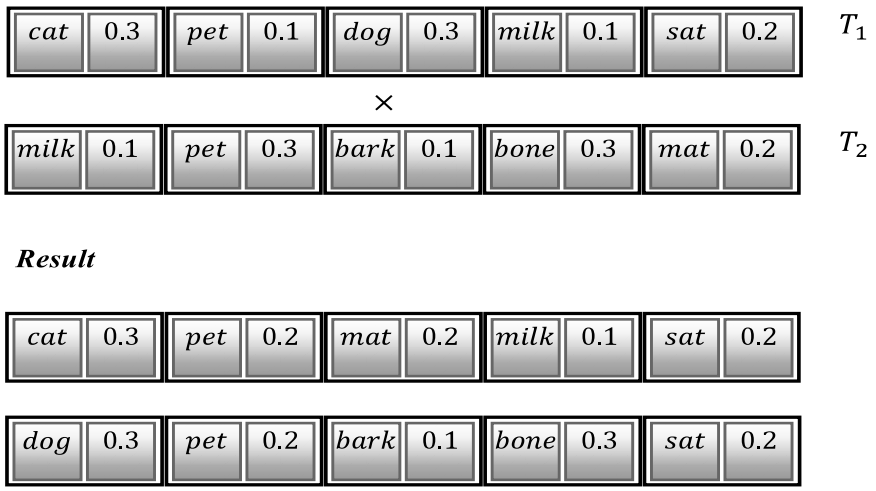


Figure 3.2: A simple two topics crossover example

3.2.4 Objectives

3.2.4.1 Coverage Score

This objective encourages topic models to represent the whole corpus. For each document, topics are evaluated by calculating the Euclidean distance between the weighted topics and the document itself. This is done by multiplying each topic's word-weight by the document's related topic weight, then calculating the distance between the resulting distribution and the document's word frequencies. Document-related topic weights are calculated using:

$$Cov(W_d, T^k) = \frac{\sum_{w \in T^k} f_{W_d,w}}{T^k_{len} - count_{w \in T^k, W_d} + 1} \tag{3.1}$$

where, $f_{W_d,w}$ gives the frequency of the word w in the document W_d and $count_{w \in T^k, W_d}$ gives the number of words that exist in the topic T^k and document W_d at the same time. Consequently, the coverage score for one document W_d can be given by:

$$Coverage(W_d) = \sqrt{\sum_{w \in W_d} \left(f_{W_d,w} - \sum_{k=1}^{K} T^k(w) Cov(W_d, T^k) \right)^2} \tag{3.2}$$

where $T^k(w)$ gives the word's weight if it is present in topic T^k, and zero otherwise. The coverage score can be normalized by its maximum value as follows:

$$nCoverage(W_d) = \sqrt{\frac{\sum_{w \in W_d} \left(f_{W_d,w} - \sum_{k=1}^{K} T^k(w) Cov(W_d, T^k) \right)^2}{\sum_{w \in W_d} f_{W_d,w}^2}}. \tag{3.3}$$

This process is repeated for all corpus documents in order to calculate a coverage score for the corpus. Eventually, there will be a vector of values which need to be minimized. The overall score for corpus W is calculated by measuring the distance between the resulting vector and the centre of the representing space using:

$$CovObj = \sqrt{\sum_{W_d \in W} nCoverage_{W_d}^2}. \tag{3.4}$$

The objective $CovObj$ needs to be minimized in MOEA-TM algorithm.

3.2.4.2 Pointwise Mutual Information Score

This objective measures the intuitive quality of a topic, in terms of how often words that co-occur in a topic tend to co-occur in general. PMI is calculated for each topic using Equation 2.62. The higher the PMI value, the more 'coherent' the topic is. For convenience, however, $1 - Coherence(T^k)$ is used as the objective, so that all objectives in MOEA-TM are to be minimized. The overall score for a topic model topics is calculated by measuring the distance between the vector of PMI scores for each topic, and the centre of the representing space using:

$$PmiObj = \sqrt{\sum_{k=1}^{K} (1 - Coherence(T^k)^2}. \tag{3.5}$$

3.2.4.3 Perplexity Score

This objective is related to the model's ability to generalize to unseen data. Strictly, the perplexity score requires a topic model which assigns a topic to every word in the entire corpus, so it cannot be calculated for topics comprising only a subset of corpus

words, which is the approach in the 'standalone' MOEA-TM. Consequently, this objective is only investigated when MOEA-TM is used to enhance a pre-calculated topic model, which is the case with the 'LDA-Initialized' MOEA-TM. The Perplexity score is calculated using the following formula:

$$PerpObj = \frac{-\sum_{\tilde{W}_d \in \tilde{W}} \log P(\tilde{W}_d|\mathcal{M})}{\sum_{\tilde{W}_d \in \tilde{W}} \tilde{N}_d} \tag{3.6}$$

where, $\mathcal{M}$ is the pre-calculated LDA topic model, $\tilde{W}$ is a small test corpus, $\tilde{W}_d$ is a document in the test corpus, and $\tilde{N}_d$ number of words in document $\tilde{W}_d$. *PerpObj* objective is calculated using Left to Right method from [152] then normalized dynamically using other calculated values. The minimized negative log-likelihood mean leads to minimized perplexity.

3.2.5 Best Solution

The primary aim is to contrast MOEA approaches to topic modelling with the standard single-objective approach, and hence a single solution is drawn from each MOEA-TM run. A compromise solution is chosen from the (approximated) Pareto front by sorting the Pareto set according to a score representing the Euclidean distance between the objective vector $\overrightarrow{v} = (v_1, v_2 \cdots v_n)$ and the centre of the objective space as follows:

$$score(\overrightarrow{v}) = \sqrt{\sum_{i=1}^{n} v_i^2}. \tag{3.7}$$

3.3 Experimental Evaluation

A number of experiments were performed to compare MOEA-TM with LDA, arguably the state-of-art in topic modelling. LDA Gibbs Sampling implementation, which is provided by the MALLET package [98], is used. MOEA implementations utilized the MOEA Framework version 1.11 [61] run by JDK version 1.6 and CentOS release 5.8.

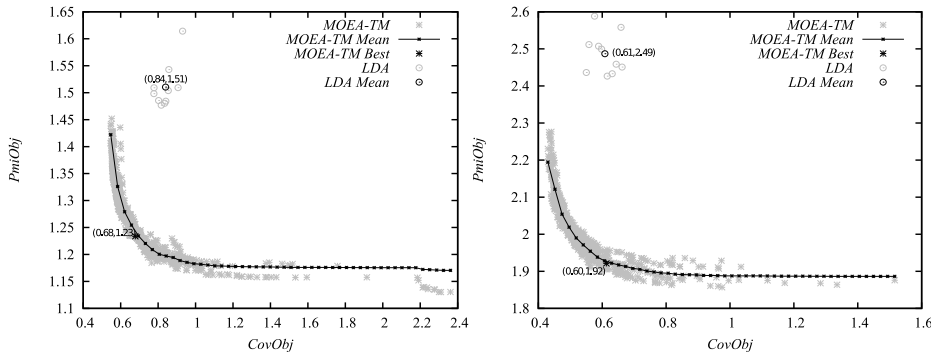


Figure 3.3: Wiki Corpus test: MOEA-TM Pareto Front and LDA solutions for ten runs (average is taken), 4 topics left and 10 topics right.

3.3.1 Corpora

The evaluation uses three corpora: the first is a very small corpus with five documents created from Wikipedia and containing four rather distinct topics (Love, Music, Sport and Government). The second corpus is made from about 15000 documents taken from news articles covering mainly four topics: Music, Economy, Fuel and Brain Surgery. The third corpus comprises about 800 documents that are summaries of projects in Information and Communication Technology (ICT) funded by the Engineering and Physical Sciences Research Council (EPSRC). Full details of each corpus are available from `http://is.gd/MOEATM`.

3.3.2 Standalone MOEA Topic Modeling

Standalone MOEA-TM was run ten times independently on each corpus, using only normalized coverage and normalized PMI objectives. LDA was also run ten times on each corpus. These experiments were done twice, once with number of topics set to 4, and once with number of topics set to 10.

Figure 3.3, Figure 3.4 and Figure 3.5 show all MOEA-TM solutions resulting from the ten runs. An averaged MOEA-TM Pareto Front is shown. The 'best' MOEA-TM solution (identified using Equation 3.7), is displayed. LDA solutions and their means are also shown. It can be seen that LDA is able to find relatively good solutions with an optimized coverage score; however, the PMI (coherence) scores are poor in comparison to those found by MOEA-TM.

Figure 3.3 and Figure 3.4 show that best MOEA-TM solution optimizes both

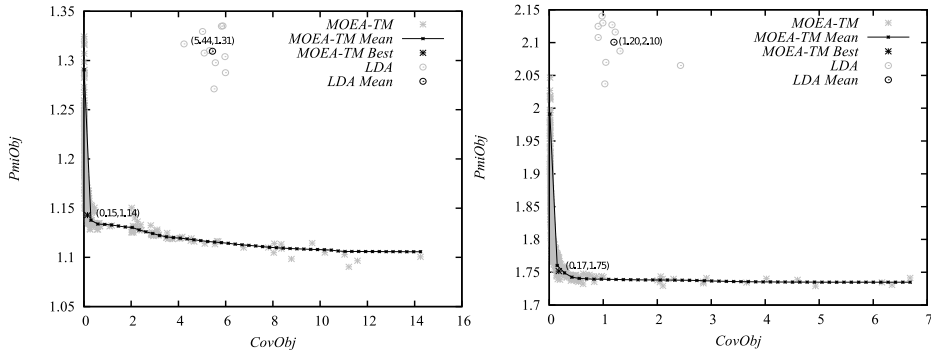


Figure 3.4: EPSRC Corpus test: MOEA-TM Pareto Front and LDA solutions for ten runs (average is taken), 4 topics left and 10 topics right.

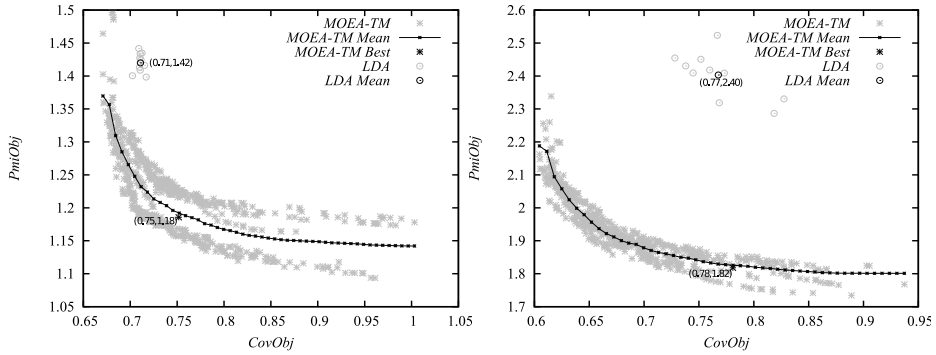


Figure 3.5: News Corpus test: MOEA-TM Pareto Front and LDA solutions for ten runs (average is taken), 4 topics left and 10 topics right.

PmiObj and *CovObj* scores for the corpora Wiki and EPSRC respectively. On the other hand, Figure 3.5 shows that for the News corpus the MOEA-TM best solution was able to optimize the *PmiObj* but not the *CovObj* objective. This means that for this corpus LDA was able to find a higher representing topics but with poor PMI.

3.3.2.1 Evaluation:

Table 3.1 and Table 3.2 show the mean and sample standard deviations of the original PMI metrics from the best MOEA-TM solutions and from LDA for 4 and 10 topic runs respectively. In these tables the higher PMI value is the better as the displayed values are the mean original normalized PMI values for solutions' topics after applying Equation 2.62 over each topic.

It can be seen that MOEA-TM outperforms LDA in terms of the PMI metric. This means that topic models resulting from MOEA-TM are significantly more coherent than topics resulting from LDA. As suggested by the standard deviations,

Table 3.1: PMI for standalone MOEA-TM and LDA, for three corpora / four topics.

	MOEA TM		**LDA**	
	Mean PMI	**SD**	**Mean PMI**	**SD**
Wiki Corpus	0.3490	0.0128	0.2460	0.0194
EPSRC Corpus	0.4119	0.0091	0.3457	0.0102
News Corpus	0.3987	0.0178	0.2933	0.0082

Table 3.2: PMI for standalone MOEA-TM and LDA for, for three corpora / ten topics.

	MOEA TM		**LDA**	
	Mean PMI	**SD**	**Mean PMI**	**SD**
Wiki Corpus	0.3483	0.0078	0.2158	0.0163
EPSRC Corpus	0.4264	0.0080	0.3371	0.0106
News Corpus	0.3913	0.0077	0.2448	0.0216

all MOEA-TM/LDA comparisons are significant with $p < 0.01$. The fact that MOEA-TM outperforms LDA in this respect is of course not very surprising, given that LDA does not directly optimize PMI; however, it is arguably surprising and interesting that the MOEA-TM approach can show such a marked improvement in topic coherence beyond that which seems achievable by LDA.

3.3.2.2 Evaluation Against A Classic Optimizer

In this section, MOEA-TM is benchmarked against a classic non-evolutionary optimizer. The exact same MOEA-TM problem representation is used for this purpose. This representation contains numbers of variables equal to $2n \cdot K$ where n is number of words inside each topic and K is total number of topics. Half of these is discrete variables to represent words, whereas the other half is continues variables to represent words' weights. Let $\mathcal{N}$ be total number of unique words in the whole corpus; consequently, possible settings for words variables are $\mathcal{N}^{n \cdot K}$ discrete states. Moreover, words' weights are continues variables, where each one might take any value between 0 and the frequency of the corresponding word in the whole corpus.

Dakota Optimization Framework The Dakota optimization toolset [1], which is developed by Sandia National Laboratories, supports a worldwide user community of scientists and engineers. It claims to deliver state-of-the-art, robust, usable software for optimization and uncertainty quantification.

Sandia National Labs indicate that computational science and engineering practitioners use Dakota across many disciplines, and they list, as follows, a number of examples where Dakota has been used to support US Department of Energy projects:

- Neutron generators performance optimization to ensure that designs meet specifications in terms of voltage, current, and space.
- Simulation models credibility establishment for thermal battery performance using a detailed verification and validation analysis.
- Sensitivity analysis of nuclear reactor fuels performance which helps to understand parameter influence in pressurized water reactors versus boiling water reactors.
- Thermal-hydraulic models parameters Calibration which simulate cooling flows within a reactor core.
- Abnormal thermal safety analysis using sparse grids, compressed sensing, and mixed aleatory-epistemic UQ methods.
- Analysis of circuit variability and performance given electrical components' radiation damage.
- Vertical axis wind turbines performance Quantification subject to uncertain gust conditions.
- Uncertain basal conditions underlying the Greenland ice sheet inference based on available observed data.
- Material performance quantification by estimation and propagation of uncertain atomistic potentials.

Asynchronous Parallel Pattern Search (APPS) The Dakota toolset is primarily oriented around continuous optimization, but has a small number of highly developed algorithms for discrete variables optimization (and hence applicable to TM), including APPS [70] [79] [81] [80].

We choose APPS to provide a comparison for the new algorithms in this chapter, in part because it is one of the few applicable 'classical' methods from the Dakota toolkit. Other reasons include the relative ease of interfacing the APPS algorithm with our TM codebase, and the fact that APPS is claimed to be particularly fast (since we need to run it multiple times to obtain a Pareto front). Also APPS has impressive performance credentials as a classical discrete optimization tool. E.g. as has been reviewed for example in [56], APPS has good credentials for better or competitive comparative performance when compared to alternatives on a range of hard real-world optimization problems such as in [57] [90] and [51].

Benchmark Wiki and EPSRC corpora where used in this benchmark. For each corpus, only $K = 4$ topics where used. Each topic contains $n = 5$ words which means that there are 40 variables in the model to be calculated. Twenty variables which are used to represent topic words whereas the rest are for representing the topic words' weights. Not all unique corpus words are fed into APPS, only the top $2K \cdot n$ high probable words in the corpus. This helps to reduce the variable space drastically and allows APPS to come up with solutions in the area where we know good solutions are located in. On the other hand, MOEA-TM explores the whole variable space. Two objectives are optimized: the coverage and the coherence of the topics. APPS is run 10 times, in each run five solutions are calculated using five random weights for each objective.

Figure 3.6 and Figure 3.7 show that APPS performance is worse than MOEA-TM on the both Wiki and ESPRC corpus respectively. LDA performs better than APPS in EPSRC corpus; whereas, in the Wiki corpus, APPS can compute solutions which are better than LDA in terms of coherence (PMI). The MOEA-TM dominates all calculated APPS solutions, this is because APPS might get stuck in local optima during the optimization process. MOEA-TM as a population-based optimization technique is able to escape local optima and converge to a global optimum.

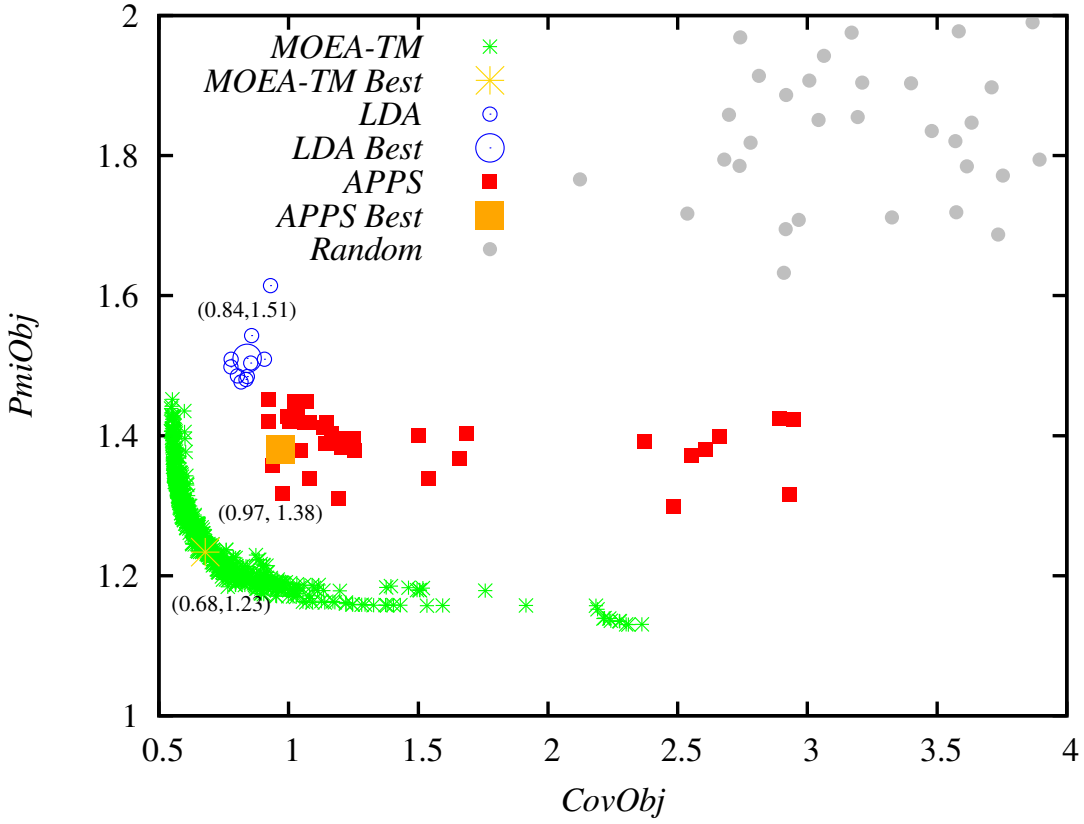


Figure 3.6: Wiki Corpus test: MOEA-TM, APPS Pareto Fronts and LDA solutions for ten runs, 4 topics.

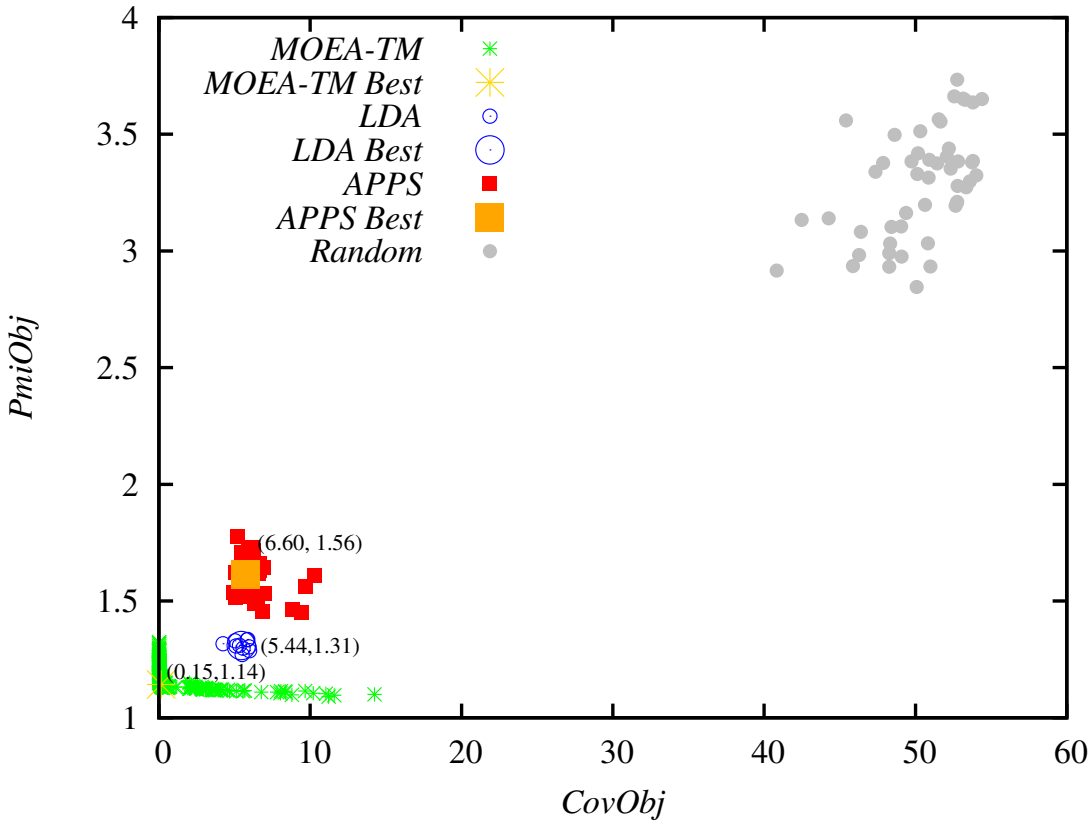
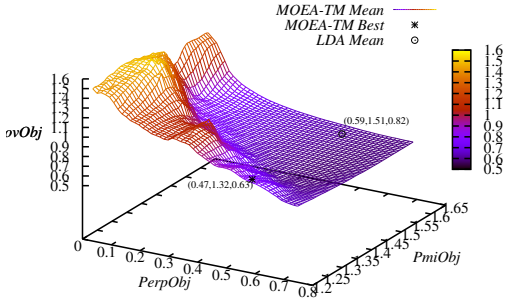
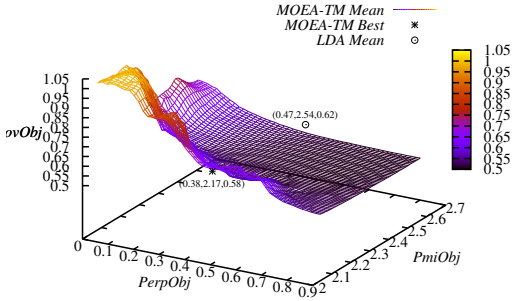


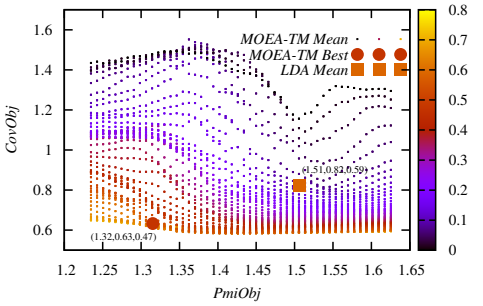
Figure 3.7: ESPRC Corpus test: MOEA-TM, APPS Pareto Fronts and LDA solutions for ten runs, 4 topics.



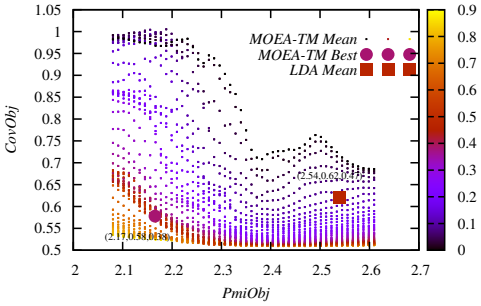
(a) 4 topics, 3D view



(b) 10 topics, 3D view



(c) 4 topics, PerpObj as color



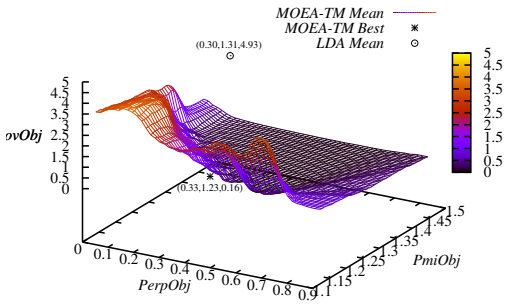
(d) 10 topics, PerpObj as color

Figure 3.8: Wiki Corpus test: LDA-Initialized MOEA-TM Pareto Front and. Pure LDA solutions for ten runs (average is taken).

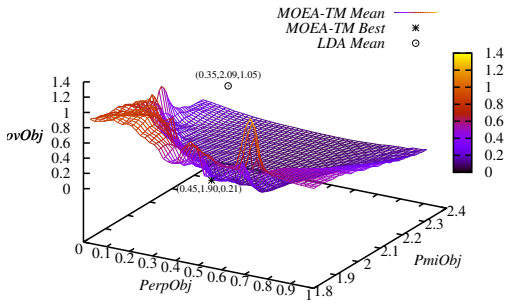
3.3.3 LDA-Initialized MOEA Topic Modelling

In this scenario, similar experiments were run but in this case, MOEA-TM is used to enhance a pre-calculated LDA topic model by optimizing three objectives *CovObj*, *PmiObj*, and *PerpObj*. The negative log-likelihood mean of an unseen test corpus words using the updated model is compared with the negative log-likelihood-mean of the same unseen test corpus words using the original LDA model. The model that has a lower negative log-likelihood mean (or higher log-likelihood mean) is better as it leads to lower perplexity. LDA-initialized MOEA-TM was run ten times and compared with (again) the results of ten un-enhanced LDA topic models.

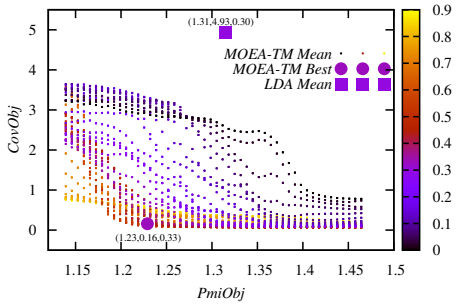
Figure 3.8, Figure 3.9 and Figure 3.10 show the average MOEA-TM Pareto Front which is calculated by interpolating all MOEA-TM Pareto Fronts and then calculating the average surface. Best MOEA-TM solution, which is identified using Equation 3.7, and LDA mean solutions are displayed in the figures. The MOEA-TM solutions and LDA solutions are not displayed for clarity. It can be seen that



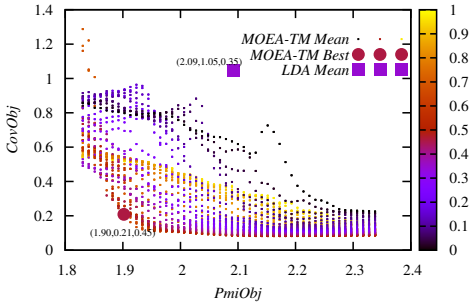
(a) 4 topics, 3D view



(b) 10 topics, 3D view



(c) 4 topics, PerpObj as color



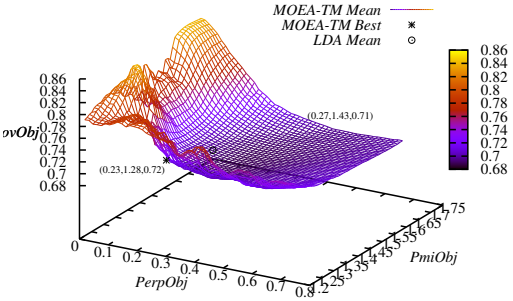
(d) 10 topics, PerpObj as color

Figure 3.9: EPSRC Corpus test: LDA-Initialized MOEA-TM Pareto Front and Pure LDA solutions for ten runs (average is taken).

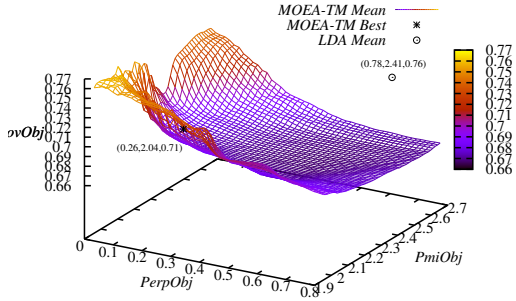
MOEA-TM was able to find better solutions in terms of Coverage ($CovObj$) and PMI ($PmiObj$) for all corpora. In terms of perplexity ($PerpObj$) Figure 3.9 shows that LDA was able to find better solutions for the EPSRC corpus. Meanwhile, MOEA-TM's best solutions have better perplexity for the Wiki and News corpora, as shown in Figure 3.8 and Figure 3.10.

3.3.3.1 Evaluation:

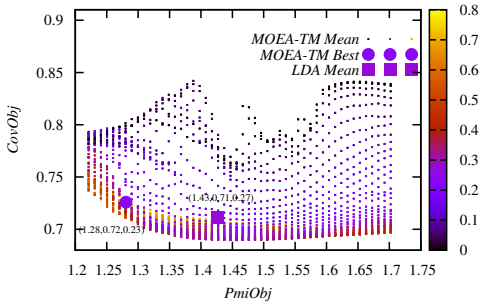
Table 3.3 and Table 3.4 present the original normalized PMI and non-normalized negative Log-Likelihood ($-LL$) metrics for LDA-Initialized MOEA-TM and LDA topic models with four and ten topics, respectively. It can be seen that LDA-Initialized MOEA-TM shows an improvement in terms of PMI values of 39%, 14% and 25% over pure LDA in the corpora Wiki, EPSRC and News, respectively when four topics are learned. When ten topics are learned the PMI improvement is 54%, 14% and 40% in the corpora Wiki, EPSRC and News, respectively. In all cases,



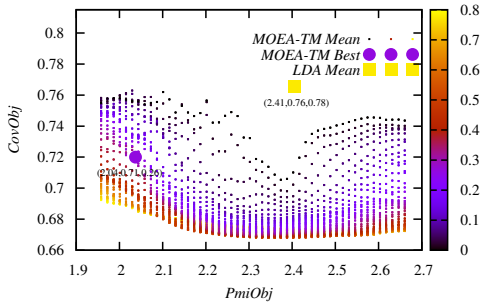
(a) 4 topics, 3D view



(b) 10 topics, 3D view



(c) 4 topics, PerpObj as color



(d) 10 topics, PerpObj as color

Figure 3.10: News Corpus test: LDA-Initialized MOEA-TM Pareto Front and Pure LDA solutions for ten runs (average is taken).

a t-test again finds that the MOEA-TM improvement in PMI is significant with $p < 0.01$, while there is, in contrast, no significance in the difference in held-out log-Likelihood values, suggests that improved coherence comes without any significant difference in the perplexity of the enhanced model.

Table 3.3: PMI scores for LDA-Initialized MOEA-TM and Pure LDA for the three corpora with four topics.

	MOEA TM			
	PMI	**SD**	**-LL**	**SD**
Wiki Corpus	0.3443	0.1129	8.1137	0.0477
EPSRC Corpus	0.3933	0.0107	8.1502	0.0074
News Corpus	0.3653	0.0069	8.7810	0.1126
	LDA			
	PMI	**SD**	**-LL**	**SD**
Wiki Corpus	0.2476	0.1932	8.1188	0.0514
EPSRC Corpus	0.3429	0.0094	8.1485	0.0062
News Corpus	0.2903	0.0142	8.8058	0.13

This is not surprising as the model is only optimizing top words in the topic model. MOEA-TM introduces limited— yet effective —changes which target objec-

Table 3.4: PMI scores for LDA-Initialized MOEA-TM and Pure LDA for the three corpora with ten topics.

	MOEA TM			
	PMI	**SD**	**-LL**	**SD**
Wiki Corpus	0.3105	0.0135	8.0716	0.0294
EPSRC Corpus	0.3889	0.0085	8.1036	0.0027
News Corpus	0.3428	0.0159	8.765	0.0896
	LDA			
	PMI	**SD**	**-LL**	**SD**
Wiki Corpus	0.2013	0.0194	8.0822	0.0262
EPSRC Corpus	0.3404	0.0101	8.1025	0.0030
News Corpus	0.2445	0.0208	8.7768	0.1162

tives that are not directly optimized by LDA such as topics' coherence. Unfortunately, extending MOEA-TM to work on all corpus words is time-consuming. Thus, for the remaining of this chapter, the focus is to optimise log-likelihood directly using a GA.

3.4 Optimizing LDA Model Log-Likelihood

It is costly to optimize perplexity and PMI metrics directly since the perplexity calculation involves iterating over test held-out documents multiple times. On the other hand, calculating the PMI score involves looking up words correlations in a Wikipedia index stored on a hard drive. It is clear that the perplexity objective could not be optimized well because the MOEA-TM algorithm is making changes on a small number of key terms only, leaving the rest of the topic model with no change. In order to design a genetic algorithm (GA) which makes more widespread changes to the whole topic model, a fast and efficient fitness function should be used. In this section, a GA is designed which will optimize the model's log-likelihood directly. The log-likelihood calculation is not as costly as calculating perplexity, thus it is practical to use it as an objective function. Consequently, 'LDA-GA' which is an LDA based single-objective optimization genetic algorithm, is elaborated next.

3.4.1 LDA-GA Design

LDA-GA is a genetic algorithm, which is designed to optimize an LDA model's topic assignments starting from a Gibbs sample. The objective is to improve the model's log-likelihood value by changing the topic assignment settings of corpus words. The algorithm is fully illustrated below.

3.4.1.1 Encoding and Initial Population

The main aim of this algorithm is to check whether further optimizing word topic assignments can lead to better performing topic models. Hence, estimated LDA models are provided as an initial population; this saves resources so the algorithm will not spend a lot of time exploring the whole solution space. Each chromosome is a full topic assignment Z, which comprises M vectors where each vector Z_j represents topic assignments for document $j \in [1, .., M]$. A topic assignment is a number $k \in [1, ..K]$ where K is the number of topics. With such long chromosomes, one needs to design fast genetic operators and a simple fitness function; otherwise the algorithm will not converge quickly.

3.4.1.2 Genetic Operators

For this proposed algorithm, a mutation operator is only used. Crossover is discarded as it is difficult to match encodings of two distinct topic models. Each topic model uses its own encoding to represent topics; thus, the same topic might not have the same numbers across different topic models. Although combinatorial optimization algorithms such as the Hungarian method [83] can be used to match topics of different models, it is costly to be used inside a genetic operator. The mutation operator works as follows: firstly, it selects a random document W_j then a random word $W_{j,t}$ from document W_j. The selected word's topic assignment is changed to match a topic assignment of another word used as a reference word from the same document.

3.4.1.3 Fitness Function

It is essential to use a fitness function which is not unduly costly to evaluate. Thus, the LDA model's log-likelihood value, which is given by the following equation, is used:

$$\log P(W, Z|\alpha, \beta) = \sum_{j=1}^{M} \log B(\widehat{z_{j,\circ}} + \alpha) - \log B(\alpha) + \sum_{k=1}^{K} \log B(\widehat{z_{\circ}^{k}} + \beta) - \log B(\beta) \ ,$$

where B is the Dirichlet normalization constant defined in Equation 2.3, α and β are the LDA model's hyper-parameters.

Simpler Fitness Function This algorithm uses only a mutation operator which changes the topic assignment for one word in the whole corpus. Consequently, the fitness function calculation can be much simpler by taking in consideration that every time only one word's topic assignment is changed and the rest of the topic assignments are kept the same. Let Z be the current topic assignments for all words in the corpus and $\tilde{Z}$ is topic assignments after applying the mutation operator. Hence, $\tilde{Z}$ is exactly the same as Z with only one topic assignment change, let the topic assignment of t^{th} word of document j ($W_{j,t} = v$) be changed from topic k to the new topic $\tilde{k}$. As a result, the difference between $\log P(W, \tilde{Z}|\alpha, \beta)$ and $\log P(W, Z|\alpha, \beta)$ is given by the following formula:

$$\log \frac{P(W, \tilde{Z}|\alpha, \beta)}{P(W, Z|\alpha, \beta)} = \log \frac{\widehat{z_{j,\circ}^{\tilde{k}}} + \alpha_{\tilde{k}}}{\widehat{z_{j,\circ}^{k}} - 1 + \alpha_k} + \log \frac{\widehat{z_{\circ,v}^{\tilde{k}}} + \beta_{\tilde{k}}}{\widehat{z_{\circ,v}^{k}} - 1 + \beta_k} - \log \frac{\widehat{z_{\circ,\circ}^{\tilde{k}}} + \beta_{\circ}}{\widehat{z_{\circ,\circ}^{k}} - 1 + \beta_{\circ}} \quad (3.8)$$

3.4.2 Experimental Results

To understand the relationship between log-likelihood (LL) and the model's ability to generalize to unseen documents, an LDA model with fixed hyperparameters alpha and beta is trained. Hence, symmetric alpha and beta are used with $\alpha_i = \frac{50}{K}$ and $\beta_r = 0.01$, where K is number of topics; the model is considered fully estimated after 1,000 iterations. Next, the log-likelihood of the previous model is optimized using LDA-GA. For each number of topics K, the experiment is repeated for ten times;

Figure 3.11 and Table 3.5 show the log-likelihood mean and standard deviation values for these multiple runs. For all cases, t-test suggests that the improvement in the log-likelihood is significant with $p < 0.001$.

Table 3.5: EPSRC corpus, Model Log-likelihood values for LDA and LDA-GA using fixed hyperparameters

	K=25		**K=50**		**K=75**		**K=150**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LDA-Mallet	-544168	446	-558083	483	-568845	361	-595408	549
LDA-GA	-518418	177	-521533	209	-525386	487	-536885	263

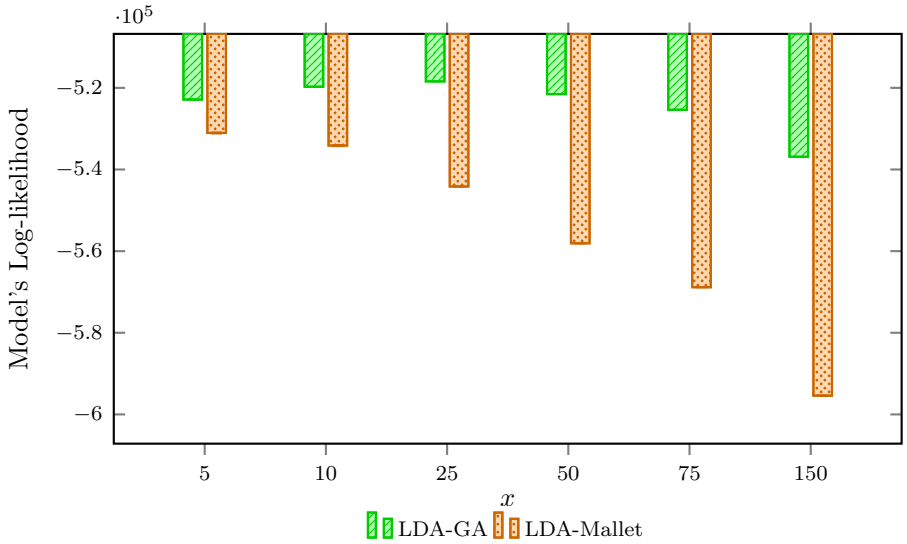


Figure 3.11: Model's log-likelihood for LDA-Mallet and LDA-GA models using fixed hyperparameters setting.

Figure 3.12 shows the perplexity scores for both LDA-Mallet and LDA-GA for different K values. Unfortunately, although LDA-GA is able to optimize the model's log-likelihood values by up to 10%, perplexity scores are not improved. In fact, perplexity scores start to deteriorate as the number of topics gets higher than 10; perplexity mean and standard deviation values are listed in Table 3.6. In these cases, t-test shows that the difference in perplexity is significant with $p < 0.001$; whereas, for the cases when the number of topics is less than or equals 10, t-test suggests that the difference is insignificant.

Table 3.6: EPSRC corpus, Held-out perplexity scores for LDA and LDA-GA with fixed hyperparameters.

	K=25		K=50		K=75		K=150	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LDA-Mallet	3149.08	6.55	3020.65	4.68	2966.55	9.82	2921.97	5.19
LDA-GA	3161.60	7.14	3048.20	6.00	2994.15	12.69	2936.62	6.32

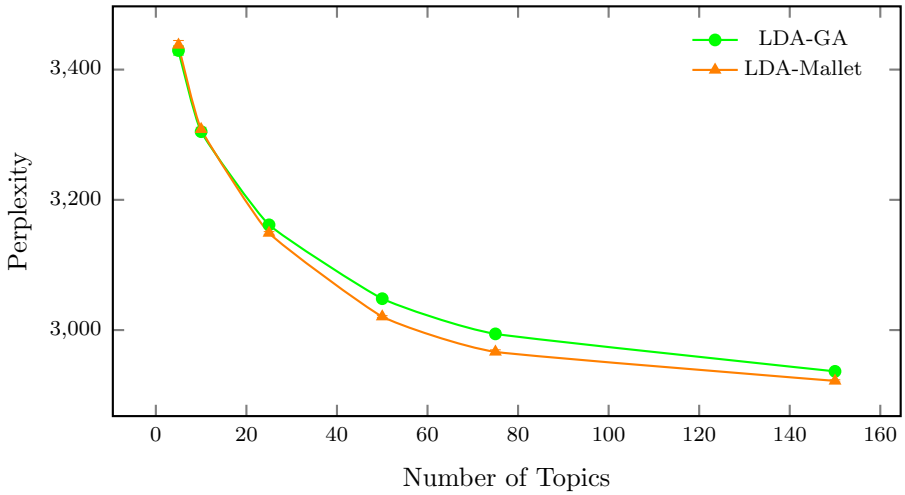


Figure 3.12: EPSRC corpus, LDA-GA, and LDA-Mallet Perplexity values for different number of topics

3.4.3 MCMC vs. Direct Optimization

In this chapter, LDA inference is done using MCMC technique; i.e. Gibbs Sampling. On the other hand, GA, which is a population-based optimization technique, was used for the same task. Generally speaking, MCMC might exhibit a poor mixing behaviour whereas optimization techniques such as expectation maximization (EM) might converge to a local optimum [133] [41] in a multimodal likelihood situation. Therefore, we use the GA to overcome the limitation of the EM. Although the GA was able to find models with higher likelihoods, MCMC was able to generalize better to unseen documents. Topic modelling is indeed a multimodal and non-concave problem [129]. Thus, direct optimization technique aims to find the mode of the multimodal distribution which is not well representing compared with MCMC method which computes an integral to find expected values of the topic model's hidden variables.

3.5 Conclusions

In this chapter, the new algorithm MOEA-TM, which shows promising performance in topic modelling, was presented. MOEA-TM initialized from LDA models is able to enhance the coherence of the topic models significantly for each of the corpora tested here. A more coherent topic model is one in which the words that tend to appear together in a topic make more sense together to a human being. This can be very useful in many topic modelling applications, such as text tagging in digital libraries, where topic coherence is particularly important [116], while in general user confidence in inferred topic models is expected to be boosted when topics are coherent. In general, multi-objective approaches may contribute significantly to topic modelling, providing the ability to specify arbitrary objectives that may be relevant in a given application, and then providing the decision maker with a diverse collection of optimal models from which the most appropriate can be selected. However, due to its extensive use of resources, MOEA-TM consumes more time compared with the original LDA. Moreover, MOEA-TM—the standalone version—limits the number of applications for which a topic model can be used. That is because it represents topics using only top words, which makes the model perform poorly in supervised tasks like classification or spam filtering. In addition, MOEA-TM—when initialized by an estimated LDA model—is not able to enhance the topic model's ability to generalize to unseen documents. This is not surprising because MOEA-TM lacks the probabilistic approach which is why it is not able to achieve a better perplexity score compared with the original LDA. In addition, further optimizing LDA model's log-likelihood leads to overfitting behaviour which means that a better model log-likelihood does not always correlate with better perplexity. The next chapters will investigate improving current probabilistic Bayesian approaches in orders to enhance both coherence and perplexity in topic models.

Chapter 4

A 'Gibbs-Newton' Technique for Enhanced Topic Models

Hyperparameters play a major role in the learning and inference process of latent Dirichlet allocation (LDA). In order to begin the LDA latent variables learning process, these parameter values need to be pre-determined. In this chapter, an extension for LDA that is called 'Latent Dirichlet allocation Gibbs Newton' (LDA-GN) is developed, LDA-GN places non-informative priors over LDA hyperparameters and uses Gibbs sampling to learn appropriate values for them. At the heart of LDA-GN is the proposed 'Gibbs-Newton' (GN) algorithm, which is a new technique for learning the parameters of multivariate Pólya distributions. In addition, a slice sampling technique for the same purpose which is called 'Gibbs slice sampling' (GSS) is proposed. The performance results of both GN and GSS is reported and compared with two prominent existing approaches to the latter task: Minka's fixed-point iteration method and the Moments method. LDA-GN is then evaluated in two ways: (i) by comparing it with both the standard LDA and LDA-GSS which is the original LDA equipped with the GSS approach, in terms of the ability of the resulting topic models to generalize to unseen documents; (ii) by comparing it with the standard LDA in its performance on a classification task.

4.1 Introduction

Most current topic modelling methods are based on the well-known 'Bag of Words' representation; in this approach, a document is simply represented as a bag of words, where words counts are preserved but their order in the original document is ignored. Latent Dirichlet allocation (LDA) [18]—the springboard for many other topic modelling methods—is the simplest topic modelling approach, and the most common one in use. However, pre-determined hyperparameters play a major role in LDA's learning and inference process; most authors, whether they use LDA or other algorithms, use fixed hyperparameter values.

In this chapter, a new extension for LDA is proposed, which removes the need to pre-determine the hyperparameters. The basic idea behind this new version of LDA, which is called 'Latent Dirichlet allocation Gibbs Newton' (LDA-GN), is to place non-informative uniform priors over the LDA hyperparameters α and β. Each component in α and β is sampled from a uniform distribution. Non-informative priors are used since prior information about these parameters is not generally available. In addition, LDA-GSS, which is the original LDA equipped with a slice sampling approach to learn its hyperparameters, is proposed.

The LDA-GN and the LDA-GSS techniques are evaluated by comparing them with the standard LDA using its recommended settings for α and β as described in [151]. This comparison is based on two evaluation metrics. Firstly, the perplexity of the inferred topic models, measured on unseen test documents (this is a common approach in the literature to evaluate topic models); secondly, the performance of LDA-GN on a supervised task such as classification is assessed using SLDA and MC-LDA.

At the heart of LDA-GN is the proposed approach 'Gibbs-Newton' (GN) for learning the parameters of a multivariate Pólya distribution; moreover, another proposed technique 'Gibbs slice sampling' (GSS), which is used for the same task, is used in LDA-GSS. However, both approaches, GN and GSS, can be extracted as standalone methods—since they are able to learn the parameters for any data distributed under a multivariate Pólya distribution—and compared with two promi-

nent methods for this task: the Moments method suggested by Ronning [130] and Minka's fixed-point iteration method [106] enhanced by Wallach [150]. A Java implementation for LDA-GN and also for standalone GN and GSS is provided at `http://is.gd/GNTMOD`.

The rest of this chapter is organised as follows: firstly, for completeness, a brief discussion of the effect of the hyperparameters in topic models is provided. Following that, the proposed GN and GSS algorithms are illustrated and evaluated by comparison with other methods in terms of accuracy and speed. Afterwards, the proposed extensions, LDA-GN and LDA-GSS, are detailed. Next, evaluation of LDA-GN and LDA-GSS is presented, before a concluding discussion.

4.2 The Effect of LDA Model Hyperparameters

The hyperparameters α and β play a large role in learning and building high-quality topic models [9, 151, 72]. Typically, symmetric values of α and β are used in the literature. Using symmetric α values means that all topics have the same chance to be assigned to a fixed number of documents. Symmetric β values mean that all terms—frequent and infrequent ones—have the same chance to be assigned to a fixed number of topics. However, according to [151], using asymmetric α and symmetric β tends to give the best performance results in terms of the inferred model's ability to generalize to unseen documents. The hyperparameters α and β generally have a smoothing effect over multinomial variables and they control the sparsity of θ and φ respectively. The sparsity of θ is controlled by α; hence smaller α values make the model prefer to describe each document using a smaller number of topics. The sparsity of φ is controlled by β; hence smaller β values makes the model reluctant to assign corresponding terms to multiple topics. Consequently, similar words with similar small β values tend to be assigned to the same subset of topics.

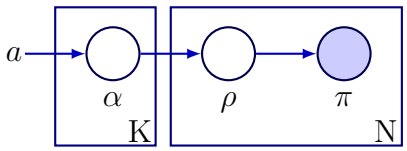


Figure 4.1: Polya distribution generative model

4.3 Estimation of Multivariate Pólya Distribution Parameters

An inspection of the LDA model reveals that the model comprises two multivariate Pólya distributions to model the data. The first distribution is used to model the distribution of the documents over topics given multinomial counts, which represents the numbers of words assigned to each topic for each document. The second distribution models the distribution of the topics over vocabulary terms, given multinomial counts of the word instances assigned to different topics in the corpus as a whole. Thus, accurate methods to learn multivariate Pólya distribution parameters can enhance the quality of LDA topic modelling at the level of documents over topics, as well as at the level of topics over vocabulary terms.

4.3.1 Bayesian Approach

The parameters of a multivariate Pólya distribution or Dirichlet distribution can be learnt from data using standard Bayesian methods. The multivariate Pólya distribution plays a major role in LDA; thus, its parameter estimation is elaborated. Given N samples from a multivariate Pólya distribution, the data can be modelled using the generative model shown in Figure 4.1. The generative process in this case amounts to first sampling a value for each of the K components from a uniform distribution with parameters $\{0,a\}$. Then, a vector ρ of dimension K is sampled from a Dirichlet distribution with parameter α. Eventually, a multinomial variable π is sampled from the multinomial distribution with parameter ρ. A non-informative uniform prior is placed before each component α_i of the parameter vector α because no prior knowledge about their values is available. The model's joint probability is:

$$P(\alpha, \pi|a) = \prod_{j=1}^{N} \int_{\rho} P(\pi|\rho)P(\rho|\alpha)d\rho \prod_{i=1}^{K} P(\alpha|a) \quad . \tag{4.1}$$

where $P(\pi|\rho) \sim Mutlnomial(\rho)$, $P(\rho|\alpha) \sim Dir(\alpha)$ and $P(\alpha|a) \sim Uniform(0, a)$. Probability densities substitution and further simplification leads to:

$$P(\alpha, \pi|a) = \frac{1}{a^K} \prod_{j=1}^{N} \frac{\Gamma(\alpha_\circ)}{\Gamma(\pi_j^\circ + \alpha_\circ)} \prod_{i=1}^{K} \frac{\Gamma(\pi_j^i + \alpha_i)}{\Gamma(\alpha_i)} \quad , \tag{4.2}$$

where, π_j^i represents the count in sample j and dimension i; whereas, π_j° represents count sum in sample j over all dimensions.

4.3.1.1 Gibbs-Newton Method

In order to learn values of the hidden variable α, a Gibbs sampler needs to be designed. The goal of Gibbs sampling here is to approximate the distribution $P(\alpha|\pi, a)$, which starts by calculating the distribution $P(\alpha_k|\alpha_{\neg k}, \pi, a)$ and then sampling each α_i value separately.

$$\begin{aligned} P(\alpha_k|\alpha_{\neg k}, \pi, a) &= \frac{P(\alpha_k, \alpha_{\neg k}, \pi|a)}{P(\alpha_{\neg k}, \pi|a)} \\ &\propto P(\alpha, \pi|a) \quad . \end{aligned} \tag{4.3}$$

It is not important to calculate the exact probability for Gibbs sampling. A ratio of probabilities is sufficient; thus, starting from the joint distribution:

$$\begin{aligned} P(\alpha_k|\alpha_{\neg k}, \pi, a) &\propto \prod_{j=1}^{N} \frac{\Gamma(\alpha_\circ)}{\Gamma(\pi_j^\circ + \alpha_\circ)} \frac{\Gamma(\pi_j^k + \alpha_k)}{a\Gamma(\alpha_k)} \prod_{i \neq k} \frac{\Gamma(\pi_j^i + \alpha_i)}{\Gamma(\alpha_i)} \\ &\propto \prod_{j=1}^{N} \frac{\Gamma(\alpha_\circ)}{\Gamma(\pi_j^\circ + \alpha_\circ)} \frac{\Gamma(\pi_j^k + \alpha_k)}{\Gamma(\alpha_k)} \quad . \end{aligned} \tag{4.4}$$

Instead of sampling from this distribution, the value which maximizes the logarithm of this density function is taken. Thus, the task is to maximize the function $\mathcal{F}(x)$ which is given by the following formula:

$$\mathcal{F}(\alpha_k) = \sum_{j=1}^{N} [\log \Gamma(\pi_j^k + \alpha_k) - \log \Gamma(\alpha_k)] - [\log \Gamma(\pi_j^\circ + \alpha_\circ) - \log \Gamma(\alpha_\circ)] \quad . \tag{4.5}$$

The first derivative of $\mathcal{F}(\alpha_k)$ is:

$$\begin{aligned}\frac{\partial[\mathcal{F}(\alpha_k)]}{\partial\alpha_k} &= \sum_{j=1}^{N}[\Psi(\pi_j^k+\alpha_k)-\Psi(\alpha_k)]-[\Psi(\pi_j^\circ+\alpha_\circ)-\Psi(\alpha_\circ)] \\ &= 0 \ .\end{aligned} \tag{4.6}$$

Unfortunately, there is no trivial solution for the previous equation; so Newton's method is used to find its root. In order to apply Newton's method, the second derivative of $\mathcal{F}(\alpha_k)$ is calculated.

$$\frac{\partial^2[\mathcal{F}(\alpha_k)]}{\partial\alpha_k^2} = \sum_{j=1}^{N}[\Psi_1(\pi_j^k+\alpha_k)-\Psi_1(\alpha_k)]-[\Psi_1(\pi_j^\circ+\alpha_\circ)-\Psi_1(\alpha_\circ)] \ . \tag{4.7}$$

where

$$\Psi_1(x) = \frac{\partial^2\log\Gamma(x)}{\partial x^2}$$

is the second derivative of the loggamma function, which is called the trigamma function [36]. It is not important to find a solution with high precision at the beginning, because it can be seen that Equation 4.6 includes the coefficient $\alpha_\circ = \sum_{k=1}^{K}\alpha_k$. This coefficient is not accurate in the first iteration of Gibbs sampling as it represents a sum of estimated values. The value of $\alpha_\circ$ is updated after each full iteration of the Gibbs sampler; in other words, after processing all α_k values. Thus, only one iteration of Newton's method is used for each α_k.

$$\alpha_k = \alpha_k^\star - \frac{\sum_{j=1}^{N}\left[\Psi(\pi_j^k+\alpha_k^\star)-\Psi(\alpha_k^\star)\right]-\left[\Psi(\pi_j^\circ+\alpha_\circ^\star)-\Psi(\alpha_\circ^\star)\right]}{\sum_{j=1}^{N}\left[\Psi_1(\pi_j^k+\alpha_k^\star)-\Psi_1(\alpha_k^\star)\right]-\left[\Psi_1(\pi_j^\circ+\alpha_\circ^\star)-\Psi_1(\alpha_\circ^\star)\right]} \ . \tag{4.8}$$

Rewriting by taking into consideration the recurrence formulae for the digamma and trigamma functions:

$$\Psi(x+n)-\Psi(x) = \sum_{l=1}^{n}\frac{1}{(x+l-1)} \tag{4.9}$$

$$\Psi_1(x+n)-\Psi_1(x) = \sum_{l=1}^{n}\frac{-1}{(x+l-1)^2} \tag{4.10}$$

where n is a positive integer $n \in \mathbb{Z}_{>0}$, gives:

$$\alpha_k = \alpha_k^\star - \frac{\sum_{j=1}^{N} \sum_{l=1}^{\pi_j^k} \frac{1}{(\alpha_k^\star + l - 1)} - \sum_{l=1}^{\pi_j^\circ} \frac{1}{(\alpha_\circ^\star + l - 1)}}{\sum_{j=1}^{N} \sum_{l=1}^{\pi_j^k} \frac{-1}{(\alpha_k^\star + l - 1)^2} - \sum_{l=1}^{\pi_j^\circ} \frac{-1}{(\alpha_\circ^\star + l - 1)^2}} \quad . \tag{4.11}$$

Rewriting using the histogram counts $\mathcal{C}$ for more computational efficiency:

$$\alpha_k = \alpha_k^\star - \frac{L_1 - \sum_{m=1}^{dim(\mathcal{C}_k)} \mathcal{C}_k^m \sum_{l=1}^{m} \frac{1}{(\alpha_k^\star + l - 1)}}{L_2 - \sum_{m=1}^{dim(\mathcal{C}_k)} \mathcal{C}_k^m \sum_{l=1}^{m} \frac{-1}{(\alpha_k^\star + l - 1)^2}} \quad . \tag{4.12}$$

where L_1 and L_2 are given by the following formulae:

$$L_1 = \sum_{m=1}^{dim(\mathcal{C}_\circ)} \mathcal{C}_\circ^m \sum_{l=1}^{m} \frac{1}{(\alpha_\circ^\star + l - 1)} \tag{4.13}$$

$$L_2 = \sum_{m=1}^{dim(\mathcal{C}_\circ)} \mathcal{C}_\circ^m \sum_{l=1}^{m} \frac{-1}{(\alpha_\circ^\star + l - 1)^2} \quad . \tag{4.14}$$

The complete GN method is described in Algorithm 9.

4.3.1.2 Slice Sampling Technique

Slice sampling [112] is a Markov chain Monte Carlo algorithm which can be used to draw samples from complex univariate and multivariate distributions. Slice sampling does not need a full knowledge about the distribution of interest; only a little information is enough. In other words, there is no need to have a prior knowledge about the normalisation constant, which is usually the intractable part of interesting Bayesian models. Thus, starting from the multivariate Pólya joint distribution given by Equation 4.2 and using Bayes rule, one can calculate the distribution of α parameter as follows:

$$P(\alpha|\pi, a) \propto \mathcal{F}(\alpha)$$

where,

$$\mathcal{F}(\alpha) = \prod_{j=1}^{N} \frac{\Gamma(\alpha_\circ)}{\Gamma(\pi_j^\circ + \alpha_\circ)} \prod_{i=1}^{K} \frac{\Gamma(\pi_j^i + \alpha_i)}{\Gamma(\alpha_i)} \tag{4.15}$$

Algorithm 9 GN method pseudo code

Input: $\mathcal{C}$ samples counts histograms, $\mathcal{C}_\circ$ samples lengths histogram.
Output: α the parameter for multivariate Pólya distribution.

```
Initialize α using Equation 2.39 and Equation 2.42 (the Moments method).
repeat
    Dgma ← 0, Tgma ← 0
    L_1 ← 0, L_2 ← 0
    α_∘ ← Σ_{i=1}^{K} α_i
    for m = 1 to dim(C_∘) do
        Dgma ← Dgma + 1/(α_∘ + m − 1)
        Tgma ← Tgma − 1/(α_∘ + m − 1)^2
        L_1 ← L_1 + C_∘^m Dgma
        L_2 ← L_2 + C_∘^m Tgma
    end for
    for i = 1 to K do
        Dgma ← 0, Tgma ← 0
        Nmtr ← 0, Dntr ← 0
        for m = 1 to dim(C_i) do
            Dgma ← Dgma + 1/(α_i + m − 1)
            Tgma ← Tgma − 1/(α_i + m − 1)^2
            Nmtr ← Nmtr + C_i^m Dgma
            Dntr ← Dntr + C_i^m Tgma
        end for
        α_i^new ← α_i − (L_1 − Nmtr)/(L_2 − Dntr)
        if α_i^new < 0 then
            α_i^new ← α_i / 2
        end if
        α_i ← α_i^new
    end for
until convergence
return α
```

Algorithm 10 shows how the logarithm of previous dense function can be calculated efficiently using count histograms.

Starting from a current sample $\alpha^\star$, the slice sampling algorithm comprises three main steps: firstly, uniformly draw a real value μ in the interval $[0, \mathcal{F}(\alpha^\star)]$ which defines a slice

$$S = \{x : \mu < \mathcal{F}(x)\} \ .$$

Then, find a hyperrectangle H around $\alpha^\star$ which contains a big part of the slice S. Eventually, uniformly draw a new point α where $\alpha \in \{H \cap S\}$. It is safer to compute $\log \mathcal{F}(\alpha^\star)$, thus the same real value y in the first step can be generated using the formula $\mu = \log \mathcal{F}(\alpha^\star) - eRand$, where $eRand$ is a random sample from

Algorithm 10 $\log \mathcal{F}(\alpha)$ evaluation

Input: $\mathcal{C}$ counts histograms, $\mathcal{C}_\circ$ lengths histogram, α.
Output: $result = \log \mathcal{F}(\alpha)$ the function evaluation at point α.

```
result ← 0
α∘ ← Σ_{i=1}^{K} α_i
for i = 1 to K do
    Lgma ← 0
    for m = 1 to dim(C_i) do
        Lgma ← Lgma + log(α_i + m − 1)
        result ← result + C_i^m Lgma
    end for
end for
Lgma ← 0
for m = 1 to dim(C∘) do
    Lgma ← Lgma + log(α∘ + m − 1)
    result ← result − C∘^m Lgma
end for
return result
```

an exponential distribution with mean 1; hence, the slice can be defined by:

$$S = \{x : \mu < \log \mathcal{F}(x)\} \quad .$$

It is important to choose a suitable hyperrectangle, because too big a hyperrectangle adversely affects the performance of the next step. On the other hand, too small a hyperrectangle hinders the algorithm from exploring the space and choosing more representative samples. Hence, an adaptive technique is used in this thesis which shrinks or expands the slice window depending on the distances of drawn samples from each other. Thus, after each sampling operation of the multivariate variable, the sampling bounds used for each component are examined and used to tell more about the distribution shape. Given all components' sampling windows, the objective is to find the maximum one and then use it as the new bound for next sampling iterations. This is defined starting from the line 32 to line 37 in Algorithm 11. In addition, the sampling process is started from an $\alpha^\star$ generated by the moments method [130]; this saves time in comparison with starting from a random point. Consequently, the algorithm used to sample α using multivariate slice sampling is illustrated in Algorithm 11.

Algorithm 11 Slice sampling technique pseudo code

Input: $\mathcal{C}$ counts histograms, $\mathcal{C}_\circ$ lengths histogram, N number of samples, *window* initial slice window, and $\mathcal{F}$ distribution proportion function.

Output: α the parameter for multivariate Pólya distribution.

```
Initialize α* using Equation 2.39 and Equation 2.42 (the Moments method).
for l = 1 to N do
  α_∘ ← Σ_{i=1}^{K} α_i
  eRand ~ Exp(1)
  μ ← log F(α*) − eRand
  for i = 1 to K do
    u ~ Uni(0, 1)
    alphaL_i ← α*_i − u · window
    alphaR_i ← alphaL_i + window
  end for
  loop
    for i = 1 to K do
      u ~ Uni(0, 1)
      α^new_i ← u · (alphaR_i − alphaL_i) + alphaL_i
    end for
    α^new_∘ ← Σ_{i=1}^{K} α^new_i
    if μ < log F(α^new) then
      break
    else
      for i = 1 to K do
        if α^new_i < α*_i then
          alphaL_i ← α^new_i
        else
          alphaR_i ← α^new_i
        end if
      end for
    end if
  end loop
  window ← 0
  for i = 1 to K do
    if window < alphaR_i − alphaL_i then
      window ← alphaR_i − alphaL_i
    end if
  end for
  α ← α + α^new, α* ← α^new
end for
return α / N
```

4.3.2 Evaluation Methodology

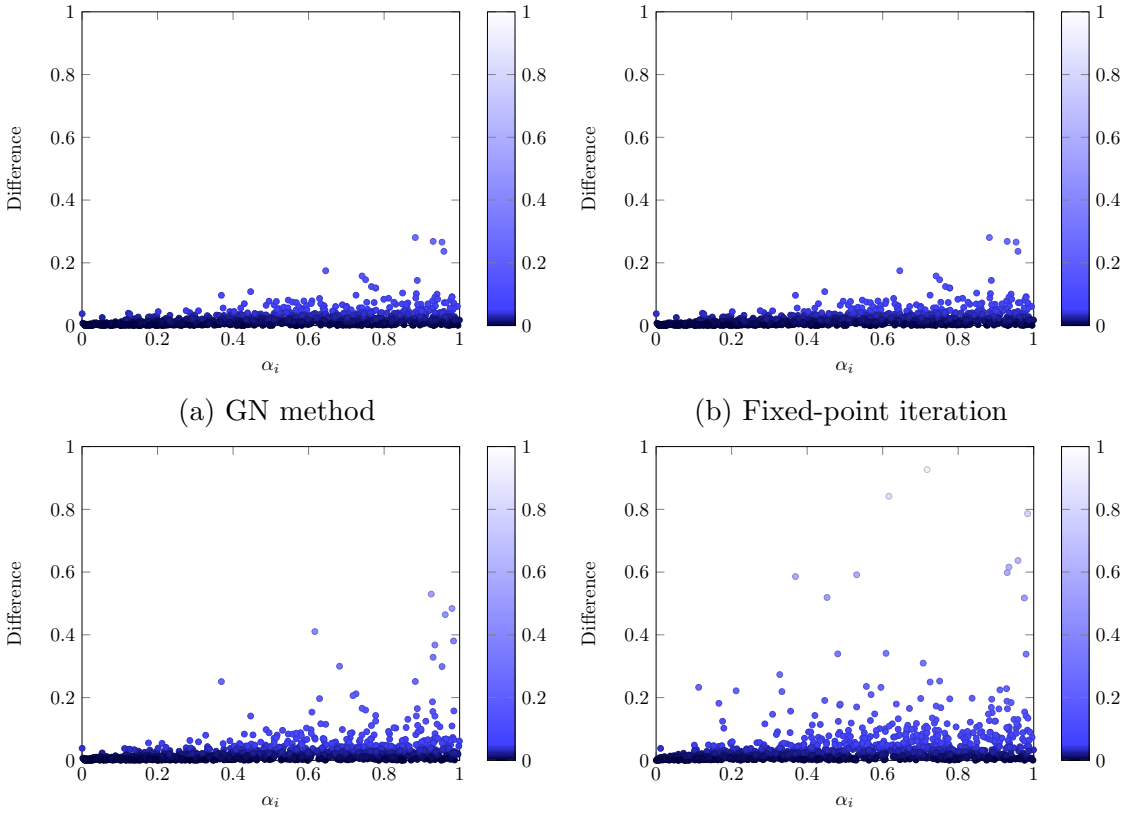
In this section, two main experiments are designed to assess the performance of the GN and GSS methods against the Moments method and Minka's fixed-point iteration. The first experiment is intended to evaluate accuracy, whereas the second experiment is aimed at assessing their efficiency. Artificial data is used, which allows to compare methods under a wide variety of conditions. The number of multivariate Pólya samples used ranges from 10 to 1000, and the number of elements used to generate each sample falls in the range $[1000, 20000]$.

4.3.2.1 Accuracy Discussion

In order to assess the accuracy of the proposed methods, two categories of data sets are considered. Both categories are designed to use a ten-dimensional multivariate Pólya distribution with known parameters α. The first category has small component values in α, being real numbers sampled uniformly from the range $]0, 1]$. The second category has relatively large α component values, in the range $]0, 50]$, again sampled uniformly. Each category can contain from 50 to 1000 multinomial count vectors or multivariate Pólya samples.

The Moments method, Minka's fixed-point iteration method, the proposed GSS technique, and the proposed GN method are used to learn the parameter α vectors from the data. Given the resulting α vector, the difference between each component α_i and its actual value is calculated and registered. 80 experiments were done, 40 for each category of data set, allowing these methods to be evaluated under highly varied settings in terms of data sparsity and number of samples needed. Figure 4.2 displays the differences of small α components and their actual values using the first set of data. Figure 4.3 shows the differences when α_i has relatively large values, in the second category of data. The figure indicates that Minka's fixed-point iteration method and the GN method record similar levels of accuracy, and both are clearly better than the Moments method in this respect. on the other hand, GSS performs better than the Moments method and worse than GN and Minka's fixed-point iteration methods. This is not surprising, as Minka's fixed-point iteration

method and the GN method are eventually maximizing the same log-likelihood function.



(a) GN method

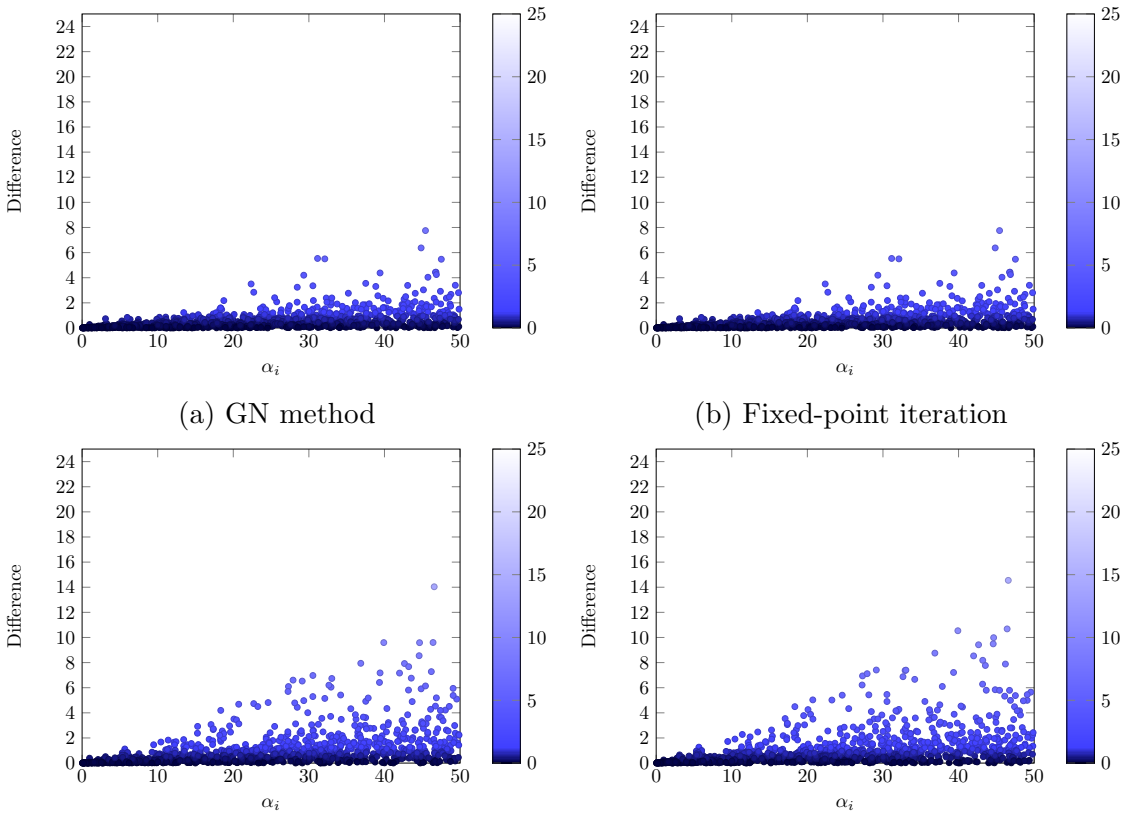
(b) Fixed-point iteration

(c) Slice sampling method

(d) Moments method

Figure 4.2: The differences between actual and learned values of α parameter components for small values of α, $\alpha_i \in]0, 1]$. The smaller the difference the better.

In [150], Wallach benchmarks Minka's fixed-point iteration method alongside other methods involving Minka's Newton iteration on the log evidence [106], fixed-point iteration on the leave-one-out log evidence [106], and fixed-point iteration on the log evidence introduced by [94]. Wallach's efficient implementation of Minka's fixed-point iteration method is the fastest and the most accurate approach [150]. Here, the proposed method is compared with Wallach's efficient implementation of Minka's fixed-point iteration method and with the Moments method [130]; the MALLET [98] implementation of the Moments method is used. It can be seen from Figure 4.2 and Figure 4.3 that the GN and Minka's fixed-point iteration methods provide the same level of accuracy. However, it is also useful to benchmark those two methods against each other in terms of speed.



(a) GN method

(b) Fixed-point iteration

(c) Slice sampling method

(d) Moments method

Figure 4.3: The differences between actual and learned values of α parameter components for large values of α, $\alpha_i \in]0, 50]$. The smaller the difference the better.

4.3.2.2 Speed Discussion

Another two data sets are generated for speed evaluation purpose. The first set is generated using a ten-dimensional multivariate Pólya distribution whereas the second set is generated using a 1000 dimensional multivariate Pólya distribution. These two datasets are used to test the performance of the proposed algorithm against Minka's fixed-point iteration method in relatively low and high dimensional cases respectively. Both distributions have a predefined parameter vector α, where all components α_i are in $]0, 1]$. For both sets, the number of multinomial counts vectors or number of samples falls in the range 10 to 1000, starting from 10 and increasing in steps of 50. The total number of elements used to generate each sample has a value between 1000 and 20000, starting at 1000 and increasing with step size 1000.

Using the first data set, and for each combination of the number of samples and number of elements, the data set is generated from the given random α values,

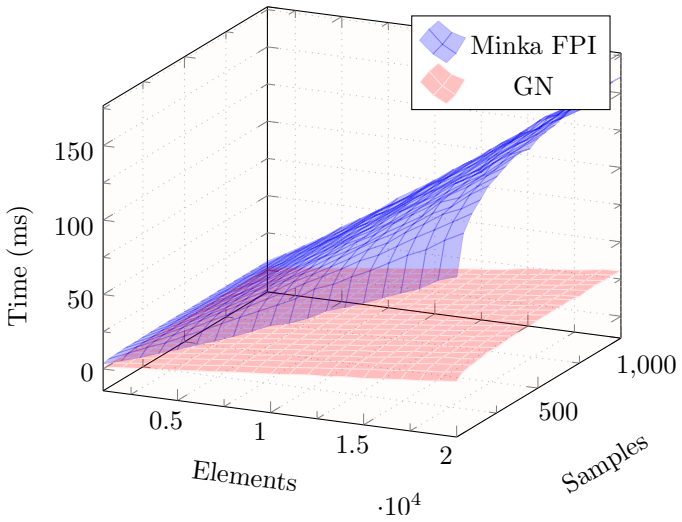


Figure 4.4: Execution time for GN and Minka's fixed-point iteration (Minka FPI) for a 10 dimensional multivariate Pólya distribution using different values of number of samples and different values of number of elements used to generate each sample

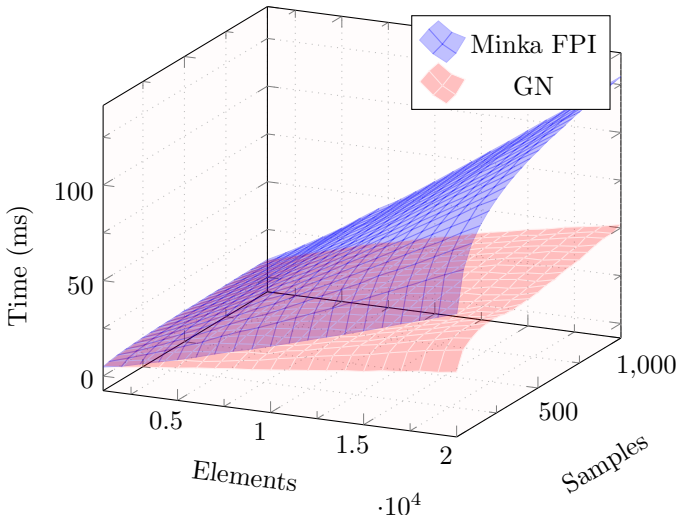


Figure 4.5: Execution time for GN and Minka's fixed-point iteration (Minka FPI) for a 1000 dimensional multivariate Pólya distribution using different values of number of samples and different values of number of elements used to generate each sample

and then the time taken by the estimation method is measured. The solution is considered to be converged (for both methods) when the maximum value among differences between previous estimates of *alpha* components values and their current estimates is less than 1.0E-6. This process is repeated 100 times, and the mean of execution time is plotted as a dot on the 3D surface is shown in Figure 4.4. The whole process was repeated 100 times, this time using the higher dimensional samples. The corresponding 3D surface for the high-dimensional trials is shown in Figure 4.5.

Figure 4.4 and Figure 4.5 show that the proposed GN method is faster than Minka's fixed-point iteration under all settings. Although the GN method requires

more computation inside each iteration over all *alpha* components values, it requires less than half the number of iterations required by Minka's fixed-point iteration method until convergence. This speed-up is more pronounced in the case of the lower-dimensional dataset; however the number of iterations needed is fewer than that required by Minka's fixed-point iteration algorithm under all settings.

4.4 LDA-GN: Incorporating Hyperparameter Inference for Enhanced Topic Models

In this section, hyperparameter estimation techniques are incorporated with the LDA, which could improve performance. Thus, the two proposed techniques are used to learn LDA hyperparameters during the learning process. Firstly, LDA-GN is proposed which is the classic LDA incorporated with Gibbs Newton technique. Then, LDA-GSS, which involves incorporating a slice sampling technique with classic LDA, is illustrated.

4.4.1 LDA-GN Model Design

LDA-GN is a variant of LDA that incorporates the proposed GN method, using it to learn variables α and β. The main idea behind LDA-GN is to allow similar words to have similar *beta* values and consequently to be distributed similarly over topics. Thus, an asymmetric *beta* prior should be used in this case. In order to learn *beta* values, the LDA model can be extended by placing a non-informative prior before *beta* variables as shown in Figure 4.6. This gives corpus words the ability to be distributed differently over topics. This is useful and necessary because some terms need to be participating in a higher number of topics compared with other terms. On the other hand, when a symmetric *beta* is used, all words have to participate in roughly the same number of topics, which can be seen as a limitation in the original LDA model. Further, it may be argued that topics should not be bounded by the number of documents that they are distributed over. Thus, an asymmetric *alpha* prior is advisable as well. The same technique is applied to *alpha*, which is, in other

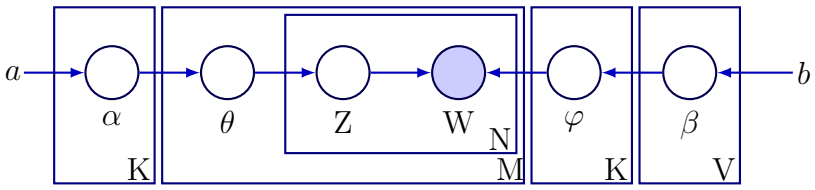


Figure 4.6: LDA-GN model

words placing a non-informative prior before the *alpha* variables, as also shown in Figure 4.6.

The generative process associated with LDA-GN is described in Algorithm 12. The LDA-GN generative process is similar to the standard LDA generative process, with an extra pair of steps. The first step is sampling each α vector component value from a uniform distribution with parameters 0 and a. The second step is sampling each β vector component value from a uniform distribution with parameters 0 and b. This will give α_k and β_v the ability to take any suitable value in the range $[0, a]$ and $[0, b]$ respectively; where a and b are a positive real numbers. The remaining steps of the generative process are the same as in the standard LDA model.

Algorithm 12 LDA-GN generative process

for $v = 1$ **to** V **do**
  Choose a beta value $\beta_v \sim Uni(0, b)$
end for
for $k = 1$ **to** K **do**
  Choose an alpha value $\alpha_k \sim Uni(0, a)$
  Choose a distribution over terms $\varphi_k \sim Dir(\beta)$
end for
for $d = 1$ **to** M **do**
  Draw a topic proportion $\theta_d \sim Dir(\alpha)$
  for $t = 1$ **to** N_d **do**
    Draw a topic assignment $z_{d,n} \sim Multi(\theta_d), z_{d,n} \in 1..K$
    Draw a word $w_{d,n} \sim Multi(\varphi_{z_{d,n}})$
  end for
end for

4.4.2 LDA-GN Model Inference

From Figure 4.6 and the LDA-GN generative process described in Algorithm 12, the joint distribution is given by the following equation:

$$P(W, Z, \theta, \varphi, \beta, \alpha | a, b) = \prod_{r=1}^{V} P(\beta_r|b) \prod_{k=1}^{K} P(\alpha_k|a) P(\varphi_k|\beta) \prod_{d=1}^{M} P(\theta_d|\alpha) \prod_{t=1}^{N_d} P(Z_{d,t}|\theta_d) P(W_{d,t}|\varphi_{Z_{d,t}}) \quad . \quad (4.16)$$

Again, the conjugacy between Dirichlet and multinomial distributions allows θ and φ to be marginalized out:

$$P(W, Z, \beta, \alpha | a, b) = \prod_{k=1}^{K} \frac{1}{a} \prod_{r=1}^{V} \frac{1}{b} \prod_{d=1}^{M} \frac{B(\widehat{z_{d,\circ}} + \alpha)}{B(\alpha)} \prod_{k=1}^{K} \frac{B(\widehat{z_{\circ}^{k}} + \beta)}{B(\beta)} \quad . \quad (4.17)$$

Gibbs sampling equations:

$$P(Z_{(d,t)} = k | Z_{\neg(d,t)}, W, \alpha, \beta, a, b) \propto (\widehat{z_{d,\circ}^{k,\neg(d,t)}} + \alpha_k) \frac{\widehat{z_{\circ,v}^{k,\neg(d,t)}} + \beta_v}{\sum_{r=1}^{V} \widehat{z_{\circ,r}^{k,\neg(d,t)}} + \beta_r} \quad (4.18)$$

$$P\left(\alpha_k | \alpha_{\neg k}, Z, W, \beta, a, b\right) \propto \prod_{d=1}^{M} \frac{\Gamma(\alpha_\circ)\Gamma(\widehat{z_{d,\circ}^{k}} + \alpha_k)}{\Gamma(\alpha_k)\Gamma(\widehat{z_{d,\circ}^{\circ}} + \alpha_\circ)} \propto \prod_{d=1}^{M} \frac{\prod_{l=0}^{\widehat{z_{d,\circ}^{k}}-1} \alpha_k + l}{\prod_{l=0}^{\widehat{z_{d,\circ}^{\circ}}-1} \alpha_\circ + l} \quad (4.19)$$

$$P\left(\beta_v | \beta_{\neg v}, Z, W, \alpha, a, b\right) \propto \prod_{k=1}^{K} \frac{\Gamma(\beta_\circ)\Gamma(\widehat{z_{\circ,v}^{k}} + \beta_v)}{\Gamma(\beta_v)\Gamma(\widehat{z_{\circ,\circ}^{k}} + \beta_\circ)} \propto \prod_{k=1}^{K} \frac{\prod_{l=0}^{\widehat{z_{\circ,v}^{k}}-1} \beta_v + l}{\prod_{l=0}^{\widehat{z_{\circ,\circ}^{k}}-1} \beta_\circ + l} \quad . \quad (4.20)$$

where $\widehat{z_{\circ,\circ}^{k}}$ is the total number of words assigned to the topic k in the whole corpus, and $\widehat{z_{d,\circ}^{\circ}}$ is the total number of words in the document W_d. Because $P(\theta_d|Z_d, \alpha)$ and $P(\varphi_k|Z, \beta)$ are samples from a Multivariate Pólya distribution, Equation 2.29 and Equation 2.30 can still be used to calculate the variables θ and φ respectively. This calculation can take place after Gibbs sampling convergence by using a good sample. Consequently, the LDA-GN collapsed Gibbs sampling algorithm is given by Algorithm 13.

Algorithm 13 LDA-GN collapsed Gibbs sampler

Input: W words of the corpus
Output: Z topic assignments, θ topics mixtures, φ topics distributions, α and β the models parameters.

Randomly initialize Z with integers $\in [1..K]$
repeat
  for $k = 1$ **to** K **do**
    $\alpha_k \leftarrow \arg\max_{\alpha_k} \left[\prod_{d=1}^{M} \frac{\Gamma(\alpha_\circ)\Gamma(\widehat{z^k_{d,\circ}}+\alpha_k)}{\Gamma(\alpha_k)\Gamma(\widehat{z^\circ_{d,\circ}}+\alpha_\circ)}\right]$
  end for
  for $v = 1$ **to** V **do**
    $\beta_v \leftarrow \arg\max_{\beta_v} \left[\prod_{k=1}^{K} \frac{\Gamma(\beta_\circ)\Gamma(\widehat{z^k_{\circ,v}}+\beta_v)}{\Gamma(\beta_v)\Gamma(\widehat{z^k_{\circ,\circ}}+\beta_\circ)}\right]$
  end for
  for $d = 1$ **to** M **do**
    for $t = 1$ **to** N_d **do**
      $v \leftarrow W_{d,t}$; $k \leftarrow Z_{d,t}$
      $\widehat{z^k_{d,\circ}} \leftarrow \widehat{z^k_{d,\circ}} - 1$; $\widehat{z^k_{\circ,v}} \leftarrow \widehat{z^k_{\circ,v}} - 1$; $\widehat{z^k_{\circ,\circ}} \leftarrow \widehat{z^k_{\circ,\circ}} - 1$;
      $k \sim (\widehat{z^k_{d,\circ}} + \alpha_k)\frac{\widehat{z^k_{\circ,v}}+\beta_v}{\widehat{z^k_{\circ,\circ}}+\beta_\circ}$
      $Z_{d,t} \leftarrow k$
      $\widehat{z^k_{d,\circ}} \leftarrow \widehat{z^k_{d,\circ}} + 1$; $\widehat{z^k_{\circ,v}} \leftarrow \widehat{z^k_{\circ,v}} + 1$; $\widehat{z^k_{\circ,\circ}} \leftarrow \widehat{z^k_{\circ,\circ}} + 1$;
    end for
  end for
until convergence
Calculate θ using Equation 2.29
Calculate φ using Equation 2.30
return Z,θ,φ,α,β

4.4.3 Evaluation Methodology

The LDA-GN model is benchmarked against the original LDA using multiple metrics. Firstly, it is compared with LDA in terms of the ability to generalize to unseen held-out documents. Then, performance on a classification task is explored, using two supervised models: SLDA and MC-LDA.

4.4.3.1 Perplexity

An LDA-GN model Gibbs sampler is implemented using Algorithm 13, and the MALLET [98] LDA implementation is used for LDA-Mallet. In addition, the slice sampling technique GSS which is described earlier is used in the LDA-GSS model. Recommended settings suggested by [151] are used for the LDA-Mallet model, which are: asymmetric Dirichlet prior over documents-over-topics distributions and a sym-

metric Dirichlet prior over topics-over-words distributions.

In order to train and evaluate these models, three corpora are used. The first corpus is EPSRC (623 documents containing 122,672 words and 13,035 vocabularies) [76] which comprises summaries of projects in Information and Communication Technology (ICT) funded by the Engineering and Physical Sciences Research Council (EPSRC). The second corpus is NewsAP (2,213 documents containing 453,462 words and 38,500 vocabularies) which is a subset of Associated Press (AP) data from the First Text Retrieval Conference (TREC-1) [65]. The third corpus is PubMed (4,155,256 documents containing 2,421,771 vocabularies and 229,742,438 words) which is a subset of PubMed articles abstracts. All standard English stop words are removed from the corpora before the application of learning or inference. Each corpus is divided into two parts: the first part is used for training, and the second part is used for evaluation purposes. The first part, which comprises 50% of corpus documents, is used to train the LDA, LDA-GSS and LDA-GN models. The remaining 50% is used to calculate perplexity scores using Equation 2.57. In order to calculate probabilities $P(\tilde{W}_j|W, Z, \alpha, \beta)$, a Java implementation of the Left-To-Right algorithm 7 is used, Thus, a better model should have a higher probability $P(\tilde{W}_j|W, Z, \alpha, \beta)$ value and consequently a lower perplexity score.

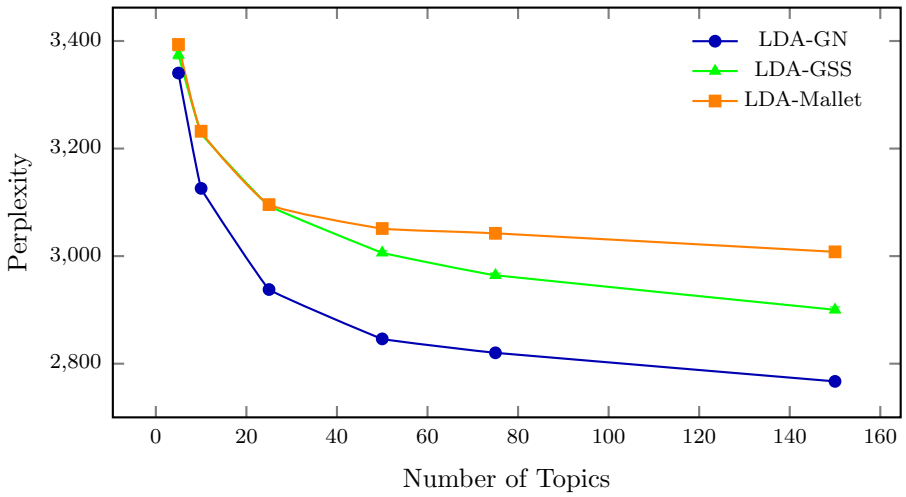


Figure 4.7: EPSRC corpus, LDA-Mallet and LDA-GN perplexity values for different number of topics

Initial values of the variable α are set as $\alpha_k = 50/K$ for all topics $k \in [1..K]$. The β variable values are initialized as $\beta_v = 0.01$ for all vocabulary terms $v \in [1..V]$.

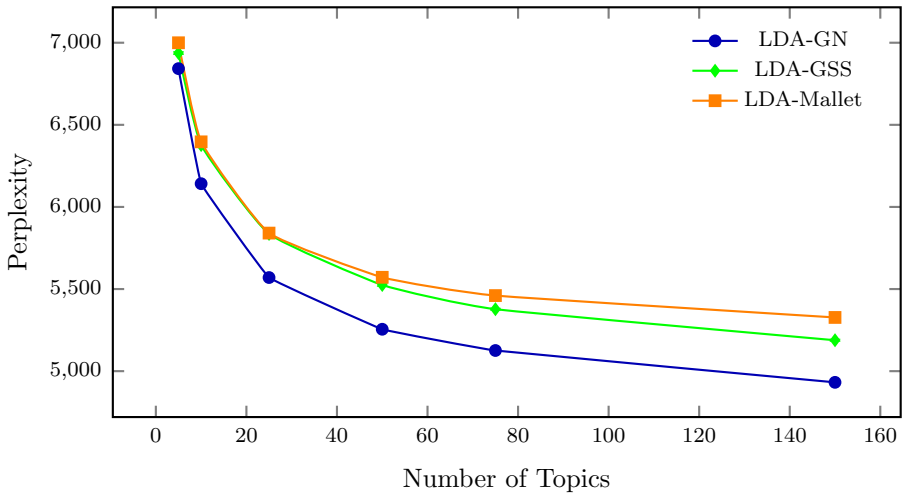


Figure 4.8: NewsAP corpus, LDA-Mallet and LDA-GN perplexity values for different number of topics

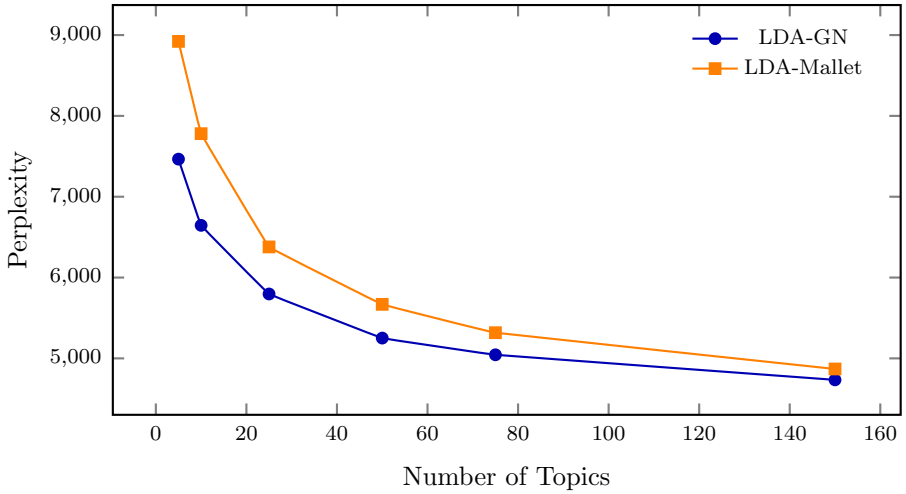
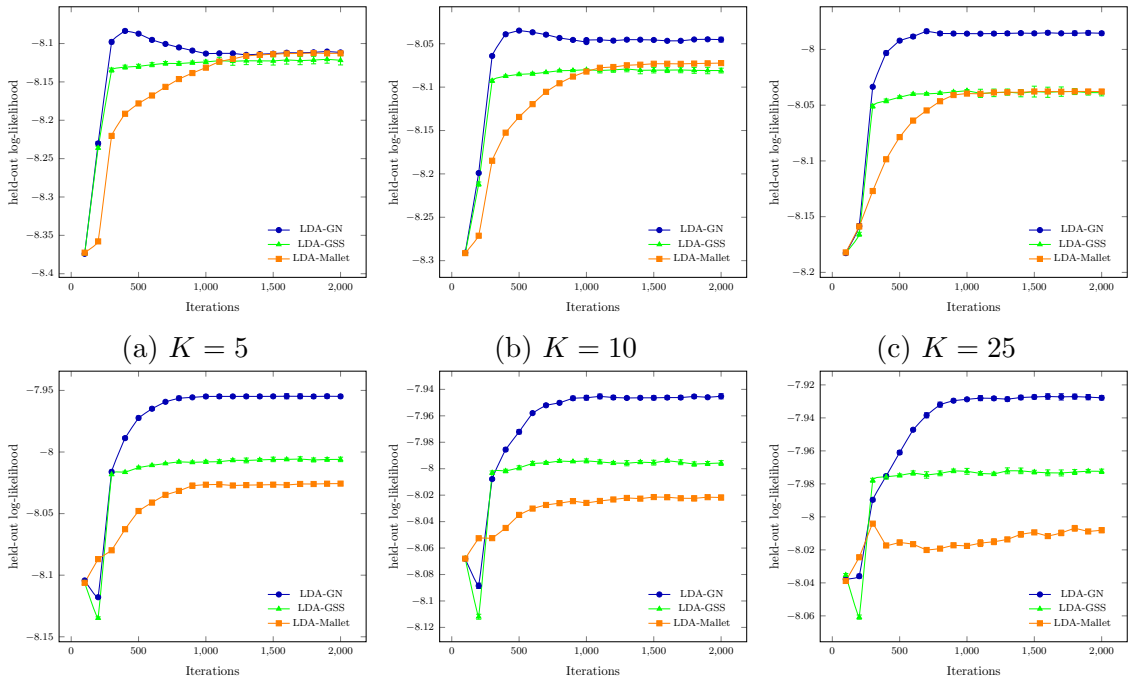


Figure 4.9: PubMed corpus, LDA-Mallet and LDA-GN perplexity values for different number of topics

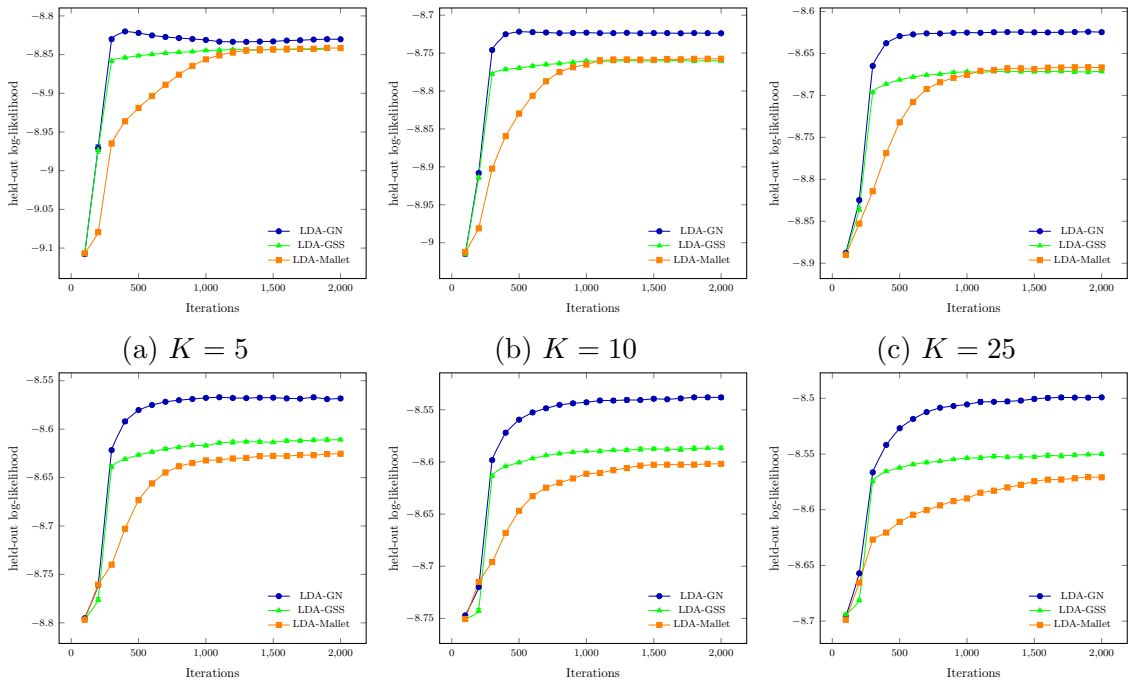
These initial values are recommended in the MALLET package documentation [98]. After that, the standard LDA model's MALLET implementation is run using a training corpus as an input. For the first 200 iterations (the burn-in period), both α and β values are kept fixed. After the burn-in period, Minka's fixed-point iteration method is used to learn α and β values from the sampler's histograms. The α and β values learning process is repeated once every 20 iterations. After 2000 iterations, the model is considered fully estimated. On the other hand, the LDA-GSS, and LDA-GN models are trained using the same training corpus which is used for standard LDA. Asymmetric α and β values are used in these models. Similarly to the standard LDA model, they are considered fully estimated after 2000 iterations.



(a) $K = 5$ (b) $K = 10$ (c) $K = 25$

(d) $K = 50$ (e) $K = 75$ (f) $K = 150$

Figure 4.10: EPSRC corpus, held-out log-likelihood scores for LDA-GN, LDA-GSS and LDA-Mallet per iteration during learning process



(a) $K = 5$ (b) $K = 10$ (c) $K = 25$

(d) $K = 50$ (e) $K = 75$ (f) $K = 150$

Figure 4.11: NewsAP corpus, held-out log-likelihood scores for LDA-GN, LDA-GSS and LDA-Mallet per iteration during learning process

The performance of LDA-Mallet, LDA-GSS and LDA-GN is tested over a range of scenarios. Each model is run ten times for each of the different settings for number of topics. For each number of topics, a fresh split is used to generate training and testing corpora. Figure 4.7, Figure 4.8 and Figure 4.9 show perplexity values of unseen test data for models inferred by LDA-GN and LDA, on the EPSRC, NewsAP and PubMed corpora respectively. Standard error bars are drawn for each point in the figures. Figure 4.7 and Figure 4.8 and Figure 4.9 show that LDA-GN outperforms standard LDA for all settings in these three corpora used for evaluation, suggesting that the topic models inferred via LDA-GN are better able to generalize than the models inferred via standard LDA (LDA-Mallet). Moreover, LDA-GSS performance is displayed in Figure 4.7 and Figure 4.8 on two corpora: EPRSC and NewsAP. LDA-GSS has a perplexity score between LDA-Mallet and LDA-GN on both corpora.

Figures 4.10 and 4.11 show held-out log-likelihood values while model is estimated for a specific number of topics K. It is clear that LDA-GN is able to converge faster than LDA-Mallet on NewAP and EPSRC corpora.

4.4.3.2 Supervised Task Performance

Another way to evaluate a topic model is to check its performance in a supervised task such as classification or spam filtering. Thus, SLDA and MC-LDA performance on classification tasks is reported next.

SLDA performance In order to benchmark LDA-GN, the SLDA model, described before in section 2.2.2, is extended to learn its parameters α and β using the GN method; this extension is called SLDA-GN. SLDA-GN's performance is benchmarked against the original SLDA, which uses Minka's fixed-point iteration method [106] to learn its parameters: symmetric β and asymmetric α.

Two corpora are used for this purpose: the Enron corpus [102], which comprises a subset of Enron emails from the period from 1999 until 2002; this corpus contains 16545 legitimate messages and 17169 spam; and the Reuters corpus, which contains 9,980 documents spread over ten categories. For both corpora, the classification task

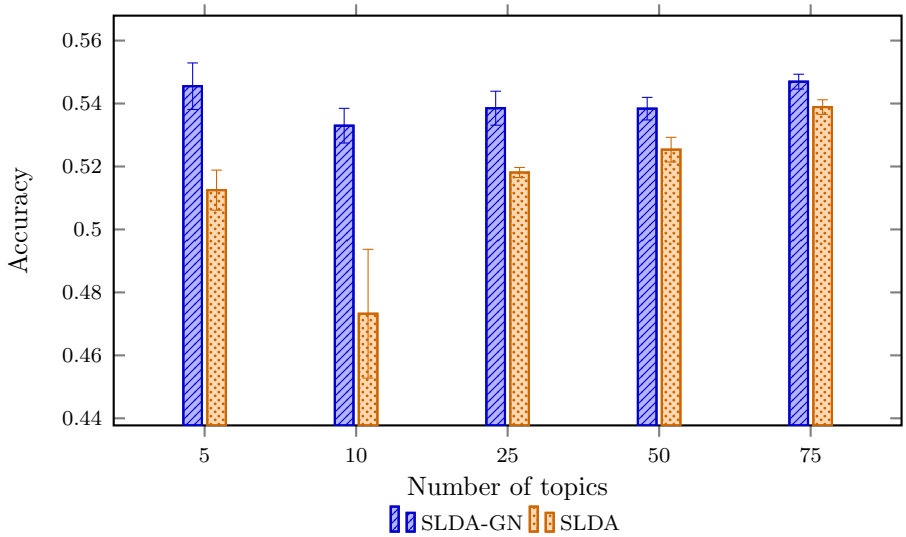


Figure 4.12: Rueters corpus with 10 classes, SLDA and SLDA-GN classification performance

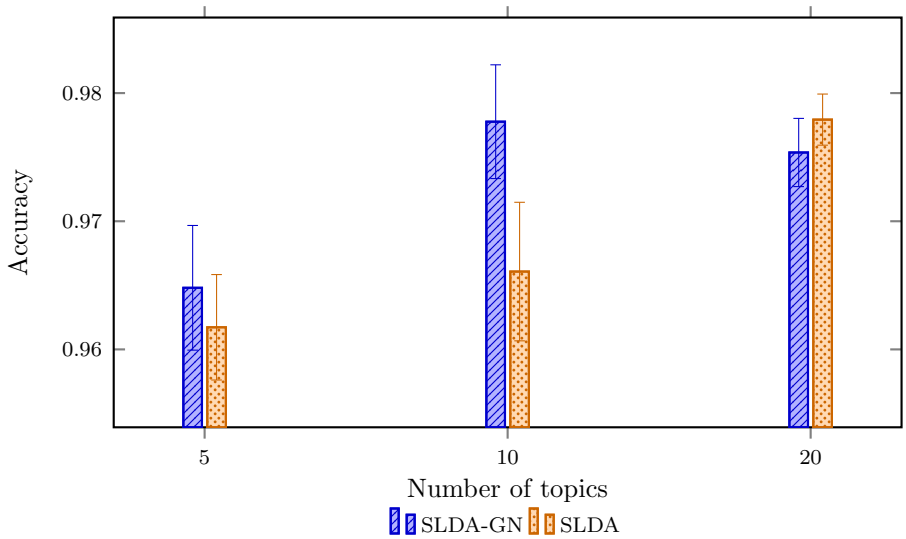


Figure 4.13: Enron corpus with 2 classes, SLDA and SLDA-GN Spam filtering performance

is run ten times for each number of topics; accuracy mean and standard deviation values are listed in Table 4.1 and Table 4.2; these results are visualised in Figure 4.12 and Figure 4.13. SLDA-GN is able to achieve higher accuracy using the same training data compared with SLDA on the Reuters corpus. However, as the number of topics gets higher, the performance gap between SLDA-GN and SLDA starts to decrease. For all different settings of topics number K, t-test shows that the difference is significant with $p < 0.05$. On the other hand, both models provide the same level of classification accuracy on Enron corpus where t-test states that difference is insignificant.

Table 4.1: Reuters classification accuracy scores for SLDA-GN and SLDA.

	K=10		K=25		K=50		K=75	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
SLDA	0.473	0.065	0.518	0.005	0.525	0.012	0.538	0.007
SLDA-GN	0.533	0.017	0.539	0.017	0.538	0.011	0.547	0.007

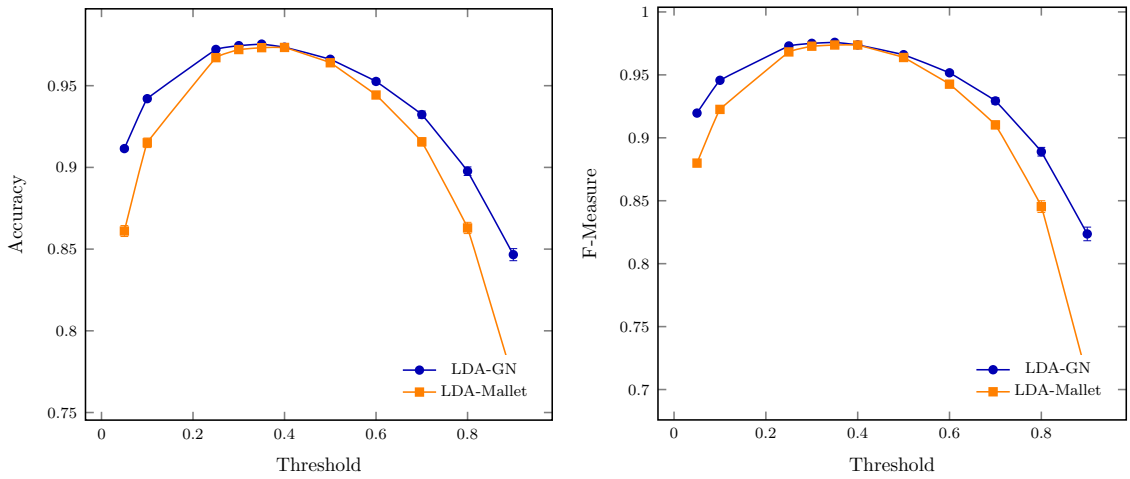
Table 4.2: Enron classification accuracy scores for SLDA-GN and SLDA.

	K=5		K=10		K=20	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
SLDA	0.9617	0.0130	0.9660	0.0170	0.9779	0.0063
SLDA-GN	0.9648	0.0154	0.9777	0.0140	0.9753	0.0084

MC-LDA Performance Two spam filters are built using the MC-LDA method elaborated in Section 2.4.3.1. The first one is built using standard LDA whereas the second one is built using LDA-GN. Three spam corpora are used for evaluation purposes: (i) the Enron Corpus;(ii) the LingSpam corpus [135] which contains 2412 legitimate message and 481 spam; (iii) The SMS Collection v.1 [5] which contains 4827 legitimate SMS messages and 747 spam SMS messages. Standard English stop words are removed from these three corpora. Each corpus is split into two parts: the first part, which comprises 80% of the corpus, is used for training whereas the remaining 20% is used for testing purposes. Using only the training part, two MC-LDA models are built using standard LDA and LDA-GN respectively.

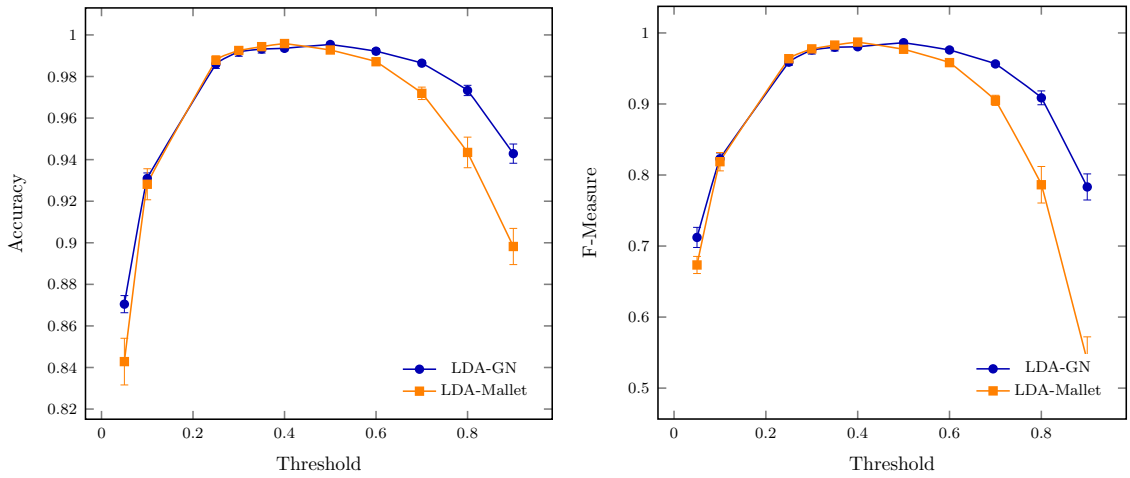
The first MC-LDA model which is built using standard LDA comprises two LDA models combined. The first one is estimated using only legitimate messages with fifty topics, whereas the second one is calculated using only spam messages with ten topics. On the other hand, a second MC-LDA model which is built using LDA-GN comprises two LDA-GN models combined. Again, the first one is calculated using legitimate messages with fifty topics, whereas the second one is estimated using spam messages with ten topics. Given two fully estimated MC-LDA models, an inference is performed for all test documents. In order to fully test the models' classification abilities, multiple thresholds are used.

For each threshold and given the trained MC-LDA models, the inference is applied three times for each model. Mean values of accuracy and f-Measure are calculated, then these points are registered in a graph. The whole process is repeated



(a) Spam filtering accuracy (b) Spam filtering f-measure

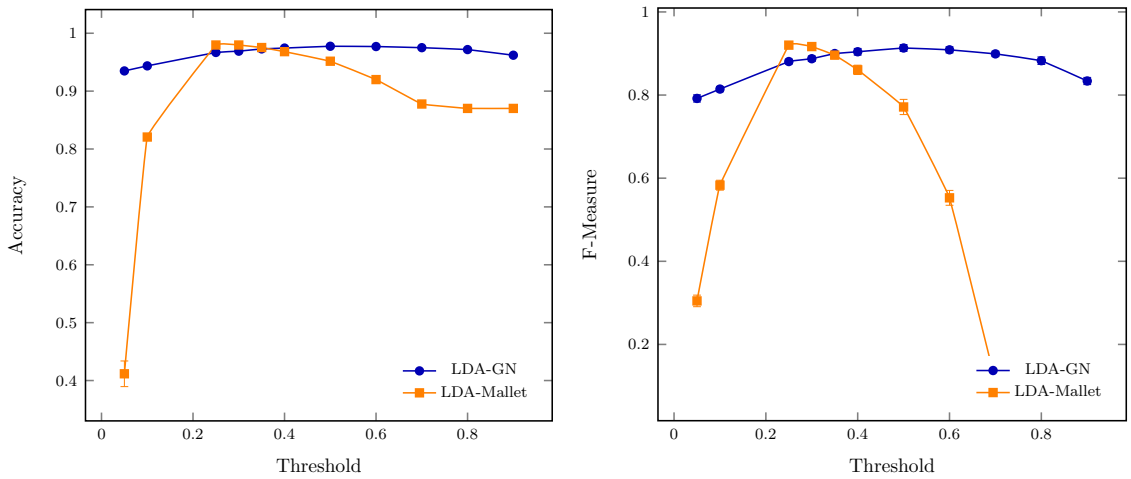
Figure 4.14: Enron corpus, LDA-Mallet and LDA-GN spam filtering performance using different threshold settings



(a) Spam filtering accuracy (b) Spam filtering f-measure

Figure 4.15: LingSpam corpus, LDA-Mallet and LDA-GN spam filtering performance using different threshold settings

five times, every time with a fresh train/test split. Eventually, the median of the five points associated with each threshold value is calculated and a curve is drawn. Figure 4.14a, Figure 4.15a and Figure 4.16a show accuracy scores for both the LDA-GN and the standard LDA models for the Enron, LingSpam and SMS Collection v.1 corpora respectively. Moreover, Figure 4.14b, Figure 4.15b and Figure 4.16b show f-Measure scores for both the LDA-GN and the standard LDA models for the Enron, LingSpam and SMS Collection v.1 corpora respectively. Perusal of these figures shows that models inferred via the LDA-GN lead to results that are less sensitive to the threshold value. However, when the right threshold value is chosen,



(a) Spam filtering accuracy

(b) Spam filtering f-measure

Figure 4.16: SMS Collection v.1 corpus, LDA and LDA-GN spam filtering performance using different threshold settings

both models are able to provide almost the same level of accuracy.

4.5 Conclusions

In this chapter, two main contributions are offered. Firstly, two new algorithms to learn multivariate Pólya distribution parameters named 'GN' and 'GSS' are described and evaluated. Secondly, based on GN and GSS, two new extensions for LDA, dubbed 'LDA-GN' and 'LDA-GSS' respectively are proposed and evaluated.

In order to assess their performance, GN and GSS are compared with two other appropriate methods: the Moments method—a quick and approximate approach—and Minka's fixed-point iteration method—a more accurate and a slower method. GN is able to infer more accurate values than the Moments method and it is able to provide the same level of accuracy provided by the Minka's fixed-point iteration method. GSS provides a level of accuracy which is between the Moments and Minka's fixed-point iteration algorithms. However, the time taken by GN to compute its results is invariably less than the time consumed by the Minka's fixed-point iteration method for the same accuracy. GN algorithm can be used in all applications that use the Dirichlet distribution or multivariate Pólya distributions to learn parameters from the data itself.

Both extensions LDA-GN and LDA-GSS show a better performance compared

with the standard LDA; however, LDA-GN is clearly better than LDA-GSS in this context. Our experiments using three corpora suggest that LDA-GN's ability to generalize to unseen documents is greater since it shows lower perplexity values over unseen documents.

Two techniques are used to measure how these models perform in a supervised task. Firstly, the SLDA model is extended to use the GN technique in the SLDA-GN model. The classification task on Reuters labelled data shows that SLDA-GN performance is better compared with the original SLDA. Secondly, the standard LDA and the LDA-GN were used in the context of the MC-LDA method in a spam classification task. Generally, LDA-GN showed better performance in this task over multiple choices of the threshold value. However, both models were able to provide the same levels of accuracy given judicious choices of the threshold value. The lower sensitivity to the threshold in the spam classification tasks—as shown by models inferred using LDA-GN—suggests that LDA-GN was able to infer higher quality topic models than LDA, being better representations and more discriminatory of the legitimate and spam parts of these corpora.

Recommended settings described in [151], which are mainly using asymmetric *alpha* and symmetric *beta* priors, lead to different words generally being constrained to contribute to the same number of topics. When a symmetric *beta* is used, and all *beta* components have a relatively large value, some words that should really only appear in a small number of topics are encouraged to spread to other topics. On the other hand, when *beta* components have a relatively small value, words tend to be distributed over a small number of topics, even though some words instances could legitimately appear in many more topics. Consequently, topic models built with these constraints can typically contain many irrelevant words among the topics. In contrast, in LDA-GN every vocabulary term has the freedom to be distributed over any number of topics with no restriction. However, with no such restriction, stop words will be encouraged to be distributed over all topics evenly. So, it is important to remove stop words before an LDA-GN model is estimated. That is why all stop words were removed in advance in LDA, LDA-GSS, and LDA-GN models.

Chapter 5

Incorporating Word Order in Topic Models

In this chapter, a new extension for LDA is proposed; it is called: 'Latent Dirichlet Allocation Correlated' (LDA-crr). In this, word correlations are represented as an observed variable based on word order. Most well-known topic models use "Bag of Words" representation for documents which cannot model the semantic relations between words. Consequently, a topic model may perform better if these relations are incorporated in the modelling process. LDA-crr is evaluated against the original LDA with fixed hyperparameter settings. Then, it is equipped with the GN and the GSS methods, and evaluated against the original LDA and the LDA-GN. Perplexity values show that the new model has a better ability to generalize to unseen documents. Also, when the number of topics gets higher, it is able to infer more coherent topics compared with the original LDA. In addition, a supervised version of the novel model, which incorporates word order in the modelling process, is proposed and evaluated against the original SLDA. Again GN and GSS are equipped with the proposed supervised model, and benchmarked against the original SLDA. When equipped with the GN approach, SLDA-crr shows the best classification performance; whereas, same level of accuracy is observed compared with the LDA-GN, when the GSS method is used. The Original SLDA, is the worst performing in the classification task.

5.1 Introduction

The majority of current topic models assume that documents' words are generated under a "Bag of Words" assumption. This is based on a matrix representation where the order of words is ignored and only word frequencies are preserved. Thus, each document is represented by a vector of term frequencies which is extremely sparse in real applications. This simple yet powerful representation allows the use of a variety of machine learning and mathematical techniques. However, a main drawback of this representation over text documents is the loss of semantic information which is vital for a human being to understand text.

In the domain of topic modelling, some topic models go beyond this representation—see for example [149, 60]—which clearly show that word orders and their semantic relations play an important role in learning higher quality models. By relaxing the "Bag of Words" assumption these methods are able to produce topic models with higher quality. However in these models, the number of parameter is expanded significantly, which consequently affects the application domain of these models. Thus, in order to keep the model simple, it should incorporate word order without adding many hidden variables to the model. The model 'Latent Dirichlet Allocation Correlated' (LDA-crr) is proposed in this chapter. LDA-crr incorporates the semantic relations between corpus words and preserves simplicity at the same time.

The remainder of this chapter is organised as follows: firstly, a discussion about term correlations in topic models is presented. After that, a technique to represent word order efficiently is elaborated. Next, the proposed model is detailed and evaluated against the LDA using fixed hyperparameter settings. Afterwards, it is equipped with the GN and the GSS methods. Eventually, the proposed model is evaluated against the LDA-GN and the original LDA in terms of perplexity, coherence, and performance in a classification task.

5.2 Term Correlations in Current Topic Models

Current topic models are based on the assumption that words are uncorrelated and are sampled independently. This is not the case in a real-world corpus; where words co-occur depending on their semantic context. This drawback is caused mainly by the usage of the "Bag of Words" representation which ignores word order in the documents. Thus, some models are proposed in the literature to address this limitation and go beyond the "Bag of Words" assumption. In [149], Wallach proposes a bi-gram topic model which is able to produce higher quality topics in terms of generalizing to unseen documents. Whereas, Griffiths et al. [60] introduce an extension for LDA which switch between LDA and a standard Hidden Markov Model (HMM) to achieve that. However, the resulting models have a much higher complexity as the number of variables expands significantly. There are other attempts in the literature to enhance a topic model's performance where word correlations are provided as prior information to the model. This approach is adopted in [114, 158, 159] where word correlations are imported from external sources and incorporated into a topic model as prior information. Although the resulting models are kept relatively simple, preparing the word correlations provides complications and require some pre-processing of the corpus words. In this chapter, a novel model is presented, which incorporates word correlations in a simple yet effective way.

5.2.1 Incorporating Term Correlations in Topic Models

Let φ be a set of K topics, which are considered K samples from a Dirichlet distribution with parameter β. According to the LDA, in order to generate a document W_d its topics distribution θ_d is sampled first from a Dirichlet distribution with parameter α. And then a topic assignment $z_{d,t}$ is sampled from a Multinomial Distribution with parameter θ_d for $W_{d,t}$, the t^{th} word of the document W_d. Eventually, the word $W_{d,t}$ itself is sampled from a Multinomial Distribution with parameter $\varphi_{z_{d,t}}$ i.e. the corresponding topic.

Thus, it is clear that topics are not correlated because the Dirichlet distribution assumes implicit independence on its proportions [16]. The Dirichlet distribution

cannot be used to detect correlations between terms directly because it cannot guarantee similar proportions over topics for semantically similar words. In [16], Blei et al. provide a topic model which supports correlations between topics. Instead of the Dirichlet distribution, they use multivariate logistic normal distribution [3] to model documents-over-topics proportions. Unlike Dirichlet distribution, multivariate logistic normal distribution is more flexible to capture correlations between proportions components.

Although this approach could work theoretically to model correlations between words in the topics-over-words proportions as well, it introduces a dramatic level of complexity to the model. This is because of the need to learn a covariance matrix of the size $V \times V$ where V is total number of unique terms in the corpus. Next, the idea of representing word order as an observed variable is illustrated.

5.3 Term Correlations as an Observed Variable

In the LDA model, each word in the corpus has one topic assignment [18] which does not depend directly on the previous word. Whereas, using word order in the learning process causes a word topic assignment to be dependent on the previous word(s) like a chain. The main idea behind this proposed model is using the previous word's topic assignments to learn more about the current word's topic.

5.3.1 The Effect of Word Order

When an author starts writing a paragraph, the transition between topics is usually smooth and sometimes the whole paragraph is talking about one topic. Thus, previous word(s) may hold valuable information to predict the current word's topic. Moreover, sometimes the semantic meaning of one word can be known only by looking at its preceding word; the next word could, for example, have a more general semantic meaning which would fit multiple topics. An example of this case would be "blood test" and "Math test", the word "test" in the first phrase is from a medical topic and in the second phrase in an education topic. In addition, phrases such as: "topic modelling", "statistical inference", form one semantic meaning; thus, all

its words should be assigned to the same topic. Such cases led other researchers to feed topic models with external correlation information in order to favour such assignment [114, 158, 159]. In the proposed model, no external information is incorporated; hence each word is influenced by the chain of the previous word(s) in the same document. Thus, these combined statistics could give the new model a better ability to learn the 'right' topic assignment for corpus words.

However, one major drawback could be caused by stop words or words which act as a stop word in the corpus. These words might introduce too much influence; consequently, words of the corpus may tend to have fewer topics. This behaviour may happen because words acting as stop words are spread over all of the corpus before a large percentage of words. These words do not hold much semantic meaning, yet they influence significantly the topic assignment for other words. Thus, the proposed model should have a mechanism to emphasise important cases only and ignore words acting as stop words, as those words can contribute to more noise to the model.

5.3.2 Representing Sequence Information

In the proposed model, the sequence information of a specific term v across all corpus documents is represented as a vector λ_v of length V. This vector shows the extent that term v should be allowed to influence successive terms. Each component $\lambda_{v,r}$ holds a binary value which is set to one if the term r is spotted immediately after v at least once in the whole corpus; otherwise it takes zero. Using the same logic for other terms, all word sequence information can be represented by a $V \times V$ asymmetric matrix λ.

For example, it is possible to represent sequence information in the following poem written by William Shakespeare using this technique:

doc1: *time is very slow for those who wait*

doc2: *very fast for those who are scared*

doc3: *very long for those who lament*

doc4: *very short for those who celebrate*

doc5: *But for those who love time is eternal*

Let the previous poem be a tiny corpus with five documents and 18 unique terms. Each term is given a numeric label as follows:
$time^{01}$, is^{02}, $very^{03}$, $slow^{04}$, for^{05}, $those^{06}$, who^{07}, $wait^{08}$, $fast^{09}$, are^{10}, $scared^{11}$, $long^{12}$, $lament^{13}$, $short^{14}$, $celebrate^{15}$, but^{16}, $love^{17}$, $eternal^{18}$.
Consequently, one can represent the whole corpus as follows:

$$W_1 = \{01, 02, 03, 04, 05, 06, 07, 08\}$$
$$W_2 = \{03, 09, 05, 06, 07, 10, 11\}$$
$$W_3 = \{03, 12, 05, 06, 07, 13\}$$
$$W_4 = \{03, 14, 05, 06, 07, 15\}$$
$$W_5 = \{16, 05, 06, 07, 17, 01, 02, 18\} \ .$$

In addition, word order information can be represented in the λ matrix presented in Figure 5.1.
The calculation of the value of Λ_v corresponding to the term v is given by the following equation:

$$\Lambda_v = \frac{\sum_{r=1}^{V} \lambda_{v,r}}{f_v} \tag{5.1}$$

where, f_v is number of times the term v appears in the whole corpus. Consequently, given a term v which is acting as a stop word, one can eliminate its influence by simply setting the corresponding value Λ_v to one. Thus, the proposed model will not favour any word when generating the word instance which follows v and acts exactly the same as in LDA. When the values of Λ vector are all ones, this model reduces to the original LDA and all word order information is ignored.

5.4 Latent Dirichlet Allocation With Correlated Words (LDA-crr)

LDA-crr is an unsupervised generative model to discover hidden topics in a collection of documents. It models not only terms frequencies in corpus of documents but also their co-occurrences.

	01	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
$\lambda_{\mathbf{1}}$	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{2}}$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$\lambda_{\mathbf{3}}$	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0
$\lambda_{\mathbf{4}}$	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{5}}$	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{6}}$	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{7}}$	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	1	0
$\lambda_{\mathbf{8}}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{9}}$	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{10}}$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
$\lambda_{\mathbf{11}}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{12}}$	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{13}}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{14}}$	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{15}}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{16}}$	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{17}}$	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\lambda_{\mathbf{18}}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5.1: Mini corpus words sequence information representation matrix λ. Terms are: $time^{01}$, is^{02}, $very^{03}$, $slow^{04}$, for^{05}, $those^{06}$, who^{07}, $wait^{08}$, $fast^{09}$, are^{10}, $scared^{11}$, $long^{12}$, $lament^{13}$, $short^{14}$, $celebrate^{15}$, but^{16}, $love^{17}$, $eternal^{18}$.

5.4.1 LDA-crr Model design

The main objective of the proposed model is to relax the "Bag of Words" assumption which is adopted by LDA and many other topic models. The proposed model LDA-crr is benchmarked against the original LDA and the LDA-GN models. Let $W = \{W_1, W_2, .., W_M\}$ be a corpus of M documents. Each document W_d comprises an ordered list or words $W_d = \{W_{d,1}, W_{d,2}, .., W_{d,N_d}\}$ where N_d is total number of words in document W_d. The variable Λ is a vector of length V where each component is corresponding to a unique term v. Each component Λ_v holds a value between zero and one, which is treated as the probability that term v topic assignment is independent of the topic assignment of the successor word. Thus, words and their order influence are used as observed variables in the model; whereas, the rest of the

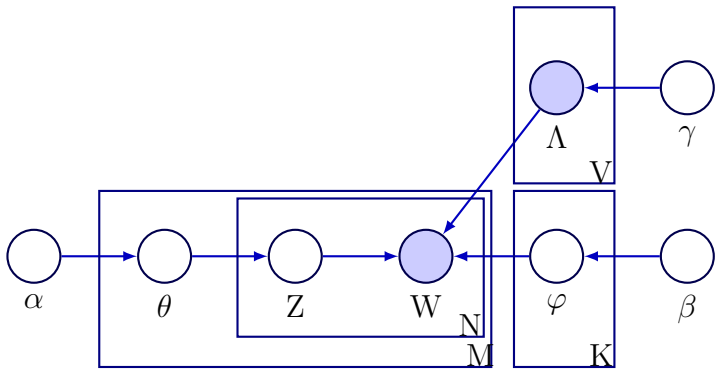


Figure 5.2: LDA-crr model

variables are hidden and need to be learnt as shown in the model's plate design in Figure 5.2. The main difference between the original LDA and the proposed model is in the way documents' words are generated. The proposed model's generative process is illustrated in Algorithm 14.

Algorithm 14 LDA-crr generative process

for $k = 1$ **to** K **do**
  Draw a topic $\varphi_k \sim Dir(\beta)$
end for
for $v = 0$ **to** V **do**
  Draw an influence probability $\Lambda_v \sim Beta(\gamma_0, \gamma_1)$
end for
for $m = 1$ **to** M **do**
  Draw a topic proportion $\theta_m \sim Dir(\alpha)$
  for $n = 1$ **to** N_m **do**
    if $n > 1$ **then**
      Draw a value $\tau \sim Bernoulli(\Lambda_{W_{m,n-1}})$
    else
      $\tau \leftarrow 1$
    end if
    if $\tau = 0$ **then**
      $Z_{m,n} \leftarrow Z_{m,n-1}$
    else
      Draw a topic assignment $Z_{m,n} \sim Multi(\theta_m),\ Z_{m,n} \in 1..K$
    end if
    Draw a word $W_{m,n} \sim Multi(\varphi_{Z_{m,n}})$
  end for
end for

5.4.2 LDA-crr Model Inference

From the LDA-crr plate representation in Figure 5.2 and its generative process in Algorithm 14, the joint probability for the LDA-crr model is given by:

$$P\left(W,\Lambda,Z,\theta,\varphi|\alpha,\beta,\gamma\right)=\prod_{v=1}^{V}P(\Lambda_v|\gamma)\cdot\prod_{i=1}^{K}P(\varphi_i|\beta)\cdot \prod_{j=1}^{M}\left(P(\theta_j|\alpha)\prod_{t=1}^{N_j}P\left(Z_{j,t}|\theta_j\right)P\left(W_{j,t}|\Lambda_{W_{j,t-1}},\varphi_{Z_{j,t}},\varphi_{Z_{j,t-1}}\right)\right) \quad (5.2)$$

The main inference problem of LDA-crr is to calculate the posterior probability given by the following equation:

$$P\left(Z|W,\Lambda,\theta,\varphi,\alpha,\beta,\gamma\right)=\frac{P\left(W,\Lambda,Z,\theta,\varphi|\alpha,\beta,\gamma\right)}{P\left(W,\Lambda|\alpha,\beta,\gamma\right)} \quad . \quad (5.3)$$

The exact posterior calculation is intractable as it involves summing over all possible settings of topic assignments Z. The inference problem becomes NP-hard for a larger number of topics [140, 64]. Fortunately, many methods are provided in the literature which can be used to approximate inference such as: Gibbs Sampling [59], Variational Inference [18], Collapsed Variational Inference [147, 108], and Collapsed Gibbs Sampling [124]. Collapsed Gibbs sampling is used to implement this model and all other models in this thesis.

5.4.2.1 LDA-crr Collapsed Gibbs Sampler

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) technique to approximate the posterior. In order to design a Gibbs sampler, full conditional distributions for all model variables need to be calculated. A collapsed Gibbs sampler [91] can be used when it is possible to integrate out some variables from the model, which makes the inference process faster. Because of the conjugacy between Dirichlet and Multinomial distributions, both θ and φ can be marginalized out from Equation 5.2,

which yields:

$$P(W, \Lambda, Z|\alpha, \beta, \gamma) = \prod_{v=1}^{V} \frac{\Lambda_v^{\gamma_0}(1-\Lambda_v)^{\gamma_1}}{B(\gamma)} \cdot \prod_{d=1}^{M} \frac{B(\widehat{z_{d,\circ}} + \alpha)}{B(\alpha)} \cdot \prod_{k=1}^{K} \frac{B(\eta_\circ^k + \beta)}{B(\beta)} \quad . \quad (5.4)$$

The value $\eta_\circ^k$ is a vector of length V, and each component value $\eta_{\circ,r}^k$ is given by the following equation:

$$\eta_{\circ,r}^k = \tilde{\Lambda}_r \widehat{z_{\circ,r}^k} + (1-\tilde{\Lambda}_r) \sum_{v=1}^{V} \widehat{z_{\circ,r}^{k,v}} \quad . \quad (5.5)$$

Where,

$$\tilde{\Lambda}_r = \frac{\sum_{v=1}^{V} \lambda_{v,r}}{f_r}$$

and $\widehat{z_{\circ,r}^{k,v}}$ represents number of terms v that immediately precede r and are assigned to topic k, and $\widehat{z_{\circ,r}^k}$ is number of terms r which are assigned to topic k in the whole corpus. $B(\alpha)$ is a multivariate version of Beta function given by the following formula:

$$B(\alpha) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K} \alpha_k)} \quad . \quad (5.6)$$

The only latent variable left in the marginal distribution is Z the words' topic assignment setting. Thus in order to calculate full conditionals, the following probabilities need to be defined:

$$\begin{aligned} P(Z_{(d,t)} = k|Z_{\neg(d,t)}, W, \Lambda, \alpha, \beta, \gamma) &= \frac{P(Z_{(d,t)} = k, Z_{\neg(d,t)}, W, \Lambda|\alpha, \beta, \gamma)}{P(Z_{\neg(d,t)}, W_{\neg(d,t)}, \Lambda|\alpha, \beta, \gamma) P(W_{(d,t)}|\alpha, \beta, \gamma)} \\ &\propto (\widehat{z_{d,\circ}^{k,\neg(d,t)}} + \alpha_k) \frac{\eta_{\circ,v}^{k,\neg(d,t)} + \beta_v}{\sum_{r=1}^{V} \eta_{\circ,r}^{k,\neg(d,t)} + \beta_r} \quad . \end{aligned} \quad (5.7)$$

Where $\eta_{\circ,r}^{k,\neg(d,t)}$ represents the value of $\eta_{\circ,r}^k$ after excluding the t^{th} word of document W_d. The values of marginalized variables θ and φ can be estimated using the current setting of topic assignments Z, which can be done using the following equations:

$$\theta_d^k = \frac{\widehat{z_{d,\circ}^k} + \alpha_k}{\sum_{i=1}^{K} \widehat{z_{d,\circ}^i} + \alpha_i} \quad (5.8)$$

$$\varphi_k^v = \frac{\widehat{z^k_{\circ,v}} + \beta_v}{\sum_{r=1}^{V} \widehat{z^k_{\circ,r}} + \beta_r} \quad . \tag{5.9}$$

Where, $\widehat{z^k_{d,\circ}}$ is number of words in document W_d which are assigned to topic k, and $\widehat{z^k_{\circ,v}}$ is number of words' instances of term v which are assigned to topic k in the whole corpus. Consequently, LDA-crr's collapsed Gibbs sampling algorithm is given by Algorithm 15, where, $\widehat{z^{k,\circ}_{\circ,v}} = \sum_{r=1}^{V} \widehat{z^{k,r}_{\circ,v}}$, $\widehat{z^k_{\circ,\circ}} = \sum_{r=1}^{V} \widehat{z^k_{\circ,r}}$ and $\beta_\circ = \sum_{r=1}^{V} \beta_r$. Although the LDA-crr model incorporates word order information and uses it to estimate words' topics assignments better, it is able to handle large corpora because the model is still simple.

Algorithm 15 LDA-crr collapsed Gibbs sampler

Input: W corpus words with order information, α and β the model's parameters.
Output: Z topic assignments, θ topics mixtures, and φ topics distributions.

Randomly initialize Z with integers $\in [1..K]$ and calculate accordantly the initial values of $\widehat{z^k_{d,\circ}}$, $\widehat{z^{k,\circ}_{\circ,v}}$, $\widehat{z^k_{\circ,v}}$ and, $\widehat{z^k_{\circ,\circ}}$
Calculate $\tilde{\Lambda}$ using corpus word order as detailed before.
repeat
  for $d = 1$ **to** M **do**
    for $t = 1$ **to** N_d **do**
      $v \leftarrow W_{d,t}$; $k \leftarrow Z_{d,t}$; $v' \leftarrow W_{d,t+1}$;
      $\widehat{z^k_{d,\circ}} \leftarrow \widehat{z^k_{d,\circ}} - 1$; $\widehat{z^{k,\circ}_{\circ,v'}} \leftarrow \widehat{z^{k,\circ}_{\circ,v'}} - 1$; $\widehat{z^k_{\circ,v}} \leftarrow \widehat{z^k_{\circ,v}} - 1$; $\widehat{z^k_{\circ,\circ}} \leftarrow \widehat{z^k_{\circ,\circ}} - 1$;
      $k \sim (\widehat{z^k_{d,\circ}} + \alpha_k) \frac{\tilde{\Lambda}_v \widehat{z^k_{\circ,v}} + (1-\tilde{\Lambda}_v)\widehat{z^{k,\circ}_{\circ,v}} + \beta_v}{\widehat{z^k_{\circ,\circ}} + \beta_\circ}$
      $Z_{d,t} \leftarrow k$
      $\widehat{z^k_{d,\circ}} \leftarrow \widehat{z^k_{d,\circ}} + 1$; $\widehat{z^{k,\circ}_{\circ,v'}} \leftarrow \widehat{z^{k,\circ}_{\circ,v'}} + 1$; $\widehat{z^k_{\circ,v}} \leftarrow \widehat{z^k_{\circ,v}} + 1$; $\widehat{z^k_{\circ,\circ}} \leftarrow \widehat{z^k_{\circ,\circ}} + 1$;
    end for
  end for
until convergence
Calculate θ using Equation 5.8
Calculate φ using Equation 5.9
return Z,θ,φ

5.4.3 Hyperparameter Estimation

Hyperparameters play a large role in learning a high-quality topic models; however, there are many methods in the literature which can be used to learn LDA-crr hyperparameters. For example: Minka's fixed point iteration method [106], the GN technique presented in section 4.3.1.1, and slice sampling [112]. In Chapter 4, the

GN method shows a better performance when it is used for LDA-GN compared with other methods such as slice sampling. In this chapter, both GN and GSS approaches are tested with the new model.

5.4.3.1 LDA-crrGN: LDA-crr with the Gibbs-Newton Technique

The GN method, which is detailed in section 4.3.1.1, is a method to learn multivariate Pólya distribution parameters by combining optimization and sampling techniques. It employs Gibbs sampling [54] and Newton optimization methods to achieve its goal. LDA-crr uses a multivariate Pólya distribution to model both documents over topics and words over topics counts.

Let $C_{\widehat{z^k_{\bullet,\circ}}}$ be a vector of frequencies, each component $C^m_{\widehat{z^k_{\bullet,\circ}}}$ represents number of times the value m is observed in all counts values $\widehat{z^k_{d,\circ}}$ for all documents $d \in [1, M]$ and topic k. In addition, let $C^m_{\widehat{z^\circ_{\bullet,\circ}}}$ be a frequency of documents which has length m. Consequently, the formula to calculate a new estimation of α_k based on current estimation $\alpha^\star_k$ is given by:

$$\alpha_k = \alpha^\star_k - \frac{\sum_{m=1}^{dim(C_{\widehat{z^\circ_{\bullet,\circ}}})} C^m_{\widehat{z^\circ_{\bullet,\circ}}} \sum_{l=1}^{m} \frac{1}{(\alpha^\star_\circ + l - 1)} - \sum_{m=1}^{dim(C_{\widehat{z^k_{\bullet,\circ}}})} C^m_{\widehat{z^k_{\bullet,\circ}}} \sum_{l=1}^{m} \frac{1}{(\alpha^\star_k + l - 1)}}{\sum_{m=1}^{dim(C_{\widehat{z^\circ_{\bullet,\circ}}})} C^m_{\widehat{z^\circ_{\bullet,\circ}}} \sum_{l=1}^{m} \frac{-1}{(\alpha^\star_\circ + l - 1)^2} - \sum_{m=1}^{dim(C_{\widehat{z^k_{\bullet,\circ}}})} C^m_{\widehat{z^k_{\bullet,\circ}}} \sum_{l=1}^{m} \frac{-1}{(\alpha^\star_k + l - 1)^2}}. \quad (5.10)$$

Similarly, the formula to calculate a new estimation of β_v based on current estimation $\beta^\star_v$ is given by:

$$\beta_v = \beta^\star_v - \frac{\sum_{m=1}^{dim(C_{\widehat{z^\bullet_{\circ,\circ}}})} C^m_{\widehat{z^\bullet_{\circ,\circ}}} \sum_{l=1}^{m} \frac{1}{(\beta^\star_\circ + l - 1)} - \sum_{m=1}^{dim(C_{\widehat{z^\bullet_{\circ,v}}})} C^m_{\widehat{z^\bullet_{\circ,v}}} \sum_{l=1}^{m} \frac{1}{(\beta^\star_v + l - 1)}}{\sum_{m=1}^{dim(C_{\widehat{z^\bullet_{\circ,\circ}}})} C^m_{\widehat{z^\bullet_{\circ,\circ}}} \sum_{l=1}^{m} \frac{-1}{(\beta^\star_\circ + l - 1)^2} - \sum_{m=1}^{dim(C_{\widehat{z^\bullet_{\circ,v}}})} C^m_{\widehat{z^\bullet_{\circ,v}}} \sum_{l=1}^{m} \frac{-1}{(\beta^\star_v + l - 1)^2}}. \quad (5.11)$$

Where, $C^m_{\widehat{z^\bullet_{\circ,\circ}}}$ is number of times the value m appeared in the counts $\widehat{z^k_{\circ,\circ}}$ for $k \in [1..K]$, and $C^m_{\widehat{z^\bullet_{\circ,v}}}$ represents number of times the value m appeared in the counts $\widehat{z^k_{\circ,v}}$ for all topics k.

5.4.3.2 LDA-crrGSS: LDA-crr with the Slice Sampling Technique

In order to learn concentration parameters, black box sampling techniques such as multivariate slice sampling [112] can be used. Multivariate slice sampling supports sampling from un-normalized probability distributions. Thus, it is enough to calculate a proportion of the following distribution:

$$\begin{aligned} P(\alpha, \beta, \gamma | W, Z, \Lambda) &\propto P(W|Z, \Lambda, \beta) P(\Lambda|\gamma) P(Z|\alpha) P(\alpha) P(\beta) P(\gamma) \\ &\propto P(W, \Lambda, Z|\alpha, \beta, \gamma) \ . \end{aligned} \tag{5.12}$$

From Equation 5.12, it could be seen that the γ value does not change when Z changes and its value depends only on the value of Λ which is an observed variable and its value does not change during the inference. Thus, it is more efficient to deal with γ separately and learn its value in the beginning of the inference process given the observed values of Λ. Let $P^*(\alpha, \beta|W, Z, \Lambda, \gamma)$ be a proportion of the distribution $P(\alpha, \beta|W, Z, \Lambda, \gamma)$.

$$P^*(\alpha, \beta|W, Z, \Lambda, \gamma) = \prod_{d=1}^{M} \frac{B(\widehat{z_{d,\circ}} + \alpha)}{B(\alpha)} \cdot \prod_{k=1}^{K} \frac{B(\mathfrak{n}_{\circ}^{k} + \beta)}{B(\beta)} \ . \tag{5.13}$$

Similarly a proportion of distribution $P(\gamma|W, Z, \Lambda, \alpha, \beta)$ is given by,

$$P^*(\gamma|W, Z, \Lambda, \alpha, \beta) = \prod_{v=1}^{V} \Lambda_v^{\gamma_0} (1 - \Lambda_v)^{\gamma_1} \ . \tag{5.14}$$

From Equation 5.13 and after using Gamma function recurrence relations [36], yields the following function:

$$\begin{aligned} &P^*(\alpha, \beta|W, Z, \Lambda, \gamma) = \\ &\qquad \frac{\prod_{k=1}^{K} \prod_{m=1}^{dim(C_{\widehat{z^{k}_{\bullet,\circ}}})} C^{m}_{\widehat{z^{k}_{\bullet,\circ}}} \mathcal{G}(\alpha_k, m)}{\prod_{m=1}^{dim(C_{\widehat{z^{\circ}_{\bullet,\circ}}})} C^{m}_{\widehat{z^{\circ}_{\bullet,\circ}}} \mathcal{G}(\alpha_{\circ,m})} \frac{\prod_{v=1}^{V} \prod_{m=1}^{dim(C_{\widehat{z^{\bullet}_{\circ,v}}})} C^{m}_{\widehat{z^{\bullet}_{\circ,v}}} \mathcal{G}(\beta_v, m)}{\prod_{m=1}^{dim(C_{\widehat{z^{\bullet}_{\circ,\circ}}})} C^{m}_{\widehat{z^{\bullet}_{\circ,\circ}}} \mathcal{G}(\beta_{\circ,m})} \ . \end{aligned} \tag{5.15}$$

Where,

$$\mathcal{G}(\zeta, m) = \frac{\Gamma(\zeta + m)}{\Gamma(\zeta)} = \prod_{l=0}^{m-1} (\zeta + l) \quad \text{where} \quad m \in \mathbb{N}_{>0}, \quad \zeta \in \mathbb{R} \ . \tag{5.16}$$

In practice, it is safer to sample from $\log P^*(\alpha, \beta | W, Z, \Lambda, \gamma)$ to avoid floating points underflow problems. Let $\hat{\alpha}$ and $\hat{\beta}$ be an estimation for α and β respectively. Multivariate slice sampling enables sampling from $\log P^*(\alpha, \beta | W, Z, \Lambda, \gamma)$ using three steps. Firstly, draw a value $y = \log P^*(\hat{\alpha}, \hat{\beta} | W, Z, \Lambda, \gamma) - eRand$ where $eRand$ is a sample from an exponential distribution with mean 1. The value y defines a slice:

$$S = \{\alpha, \beta : \log P^*(\alpha, \beta | W, Z, \Lambda, \gamma) > y\} \ .$$

Then, a hyperrectangle around the current estimations $\hat{\alpha}$ and $\hat{\beta}$ need to be constructed using a predefined window size. Finally, draw a sample from the slice S and within the hyper-rectangle. A detailed slice sampling technique algorithm is presented in Algorithm 11.

5.5 Supervised LDA-crr

Supervised LDA-crr (SLDA-crr), which is a supervised extension to LDA-crr, is proposed in this section. SLDA-crr applies same ideas used to create LDA-crr into the original SLDA model which is fully described in section 2.2.2.1; hence, the observed variable Λ is added to the model. From the plate representation shown in Figure 5.3, SLDA-crr joint probability is given by the following formula:

$$P(W, \Lambda, Y, Z, \theta, \varphi | \alpha, \beta, \gamma, \mu, \delta) = \prod_{v=1}^{V} P(\Lambda_v | \gamma) \prod_{k=1}^{K} P(\varphi_k | \beta) \cdot$$
$$\prod_{d=1}^{M} \left(P(Y_d | \overline{Z}_d, \mu, \delta) P(\theta_d | \alpha) \prod_{t=1}^{N_d} P\left(Z_{d,t} | \theta_d\right) P\left(W_{d,t} | \Lambda_{W_{d,t-1}}, \varphi_{Z_{d,t}}, \varphi_{Z_{d,t-1}}\right) \right) \tag{5.17}$$

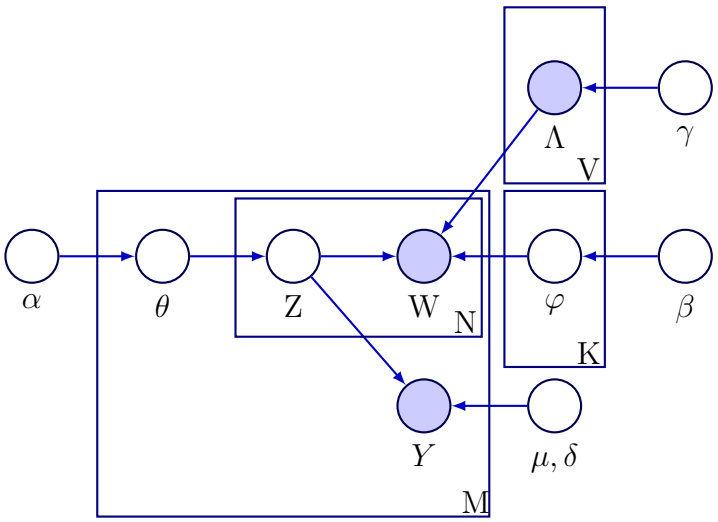


Figure 5.3: SLDA-crr model

Which yields after marginalizing both θ and φ variables:

$$P(W,\Lambda,Y,Z|\alpha,\beta,\gamma,\mu,\delta) = \prod_{v=1}^{V} \frac{\Lambda_v^{\gamma_0}(1-\Lambda_v)^{\gamma_1}}{B(\gamma)} \cdot \prod_{k=1}^{K} \frac{B(\eta_\circ^k+\beta)}{B(\beta)} \cdot \prod_{d=1}^{M} \frac{B(\widehat{z_{d,\circ}}+\alpha)}{B(\alpha)} P(Y_d|\overline{Z_d},\mu,\delta) \quad (5.18)$$

The only latent variable in the marginalized distribution is Z the topic assignment. Thus, in order to design a collapsed Gibbs sampler, a full conditional distribution needs to be defined as follows:

$$P(Z_{d,t}|Z_{\neg(d,t)},W,Y,\alpha,\beta,\mu,\delta) \propto (\widehat{z_{d,\circ}^{k,\neg(d,t)}}+\alpha_k) \cdot \frac{\eta_{\circ,v}^{k\neg(d,t)}+\beta_v}{\sum_{r=1}^{V}\eta_{\circ,r}^{k,\neg(d,t)}+\beta_r} \cdot \exp\left(\frac{\mu_k}{\delta N_d}\left(Y_d - \overline{Z_{d,\neg t}}\cdot\mu - \frac{\mu_k}{2N_d}\right)\right) \quad . \quad (5.19)$$

SLDA-crr's collapsed Gibbs sampling algorithm is given by Algorithm 16

5.6 Evaluation Methodology

In this section, the performance of LDA-crr is benchmarked against LDA and LDA-GN; firstly, the model's ability to generalize to unseen documents is measured. Then topic coherence is calculated using the normalized PMI score, which was explained in section 2.4.2. Two corpora are used for this purpose:

Algorithm 16 SLDA-crr collapsed Gibbs sampler

Input: W words of the corpus, Y documents' response values, α, β, μ and δ the model parameters.

Output: Z topic assignments, θ topics mixtures, and φ topics distributions.

Randomly initialize Z with integers $\in [1..K]$ and calculate accordantly the initial values of $\widehat{z^k_{d,\circ}}$, $\widehat{z^{k,\circ}_{\circ,v}}$, $\widehat{z^k_{\circ,v}}$ and, $\widehat{z^k_{\circ,\circ}}$

Calculate $\tilde{\Lambda}$ using corpus word order as detailed before.

repeat

  for $d = 1$ **to** M **do**

    for $t = 1$ **to** N_d **do**

      $v \leftarrow W_{d,t}$; $k \leftarrow Z_{d,t}$; $v' \leftarrow W_{d,t+1}$;

      $\widehat{z^k_{d,\circ}} \leftarrow \widehat{z^k_{d,\circ}} - 1$; $\widehat{z^{k,\circ}_{\circ,v'}} \leftarrow \widehat{z^{k,\circ}_{\circ,v'}} - 1$; $\widehat{z^k_{\circ,v}} \leftarrow \widehat{z^k_{\circ,v}} - 1$; $\widehat{z^k_{\circ,\circ}} \leftarrow \widehat{z^k_{\circ,\circ}} - 1$;

      $k \sim (\widehat{z^k_{d,\circ}} + \alpha_k) \frac{\tilde{\Lambda}_v \widehat{z^k_{\circ,v}} + (1-\tilde{\Lambda}_v)\widehat{z^{k,\circ}_{\circ,v}} + \beta_v}{\widehat{z^k_{\circ,\circ}} + \beta_\circ} \exp(\frac{\mu_k}{\delta N_d}(Y_d - \overline{Z_{d,\neg t}} \cdot \mu - \frac{\mu_k}{2N_d}))$

      $Z_{d,t} \leftarrow k$

      $\widehat{z^k_{d,\circ}} \leftarrow \widehat{z^k_{d,\circ}} + 1$; $\widehat{z^{k,\circ}_{\circ,v'}} \leftarrow \widehat{z^{k,\circ}_{\circ,v'}} + 1$; $\widehat{z^k_{\circ,v}} \leftarrow \widehat{z^k_{\circ,v}} + 1$; $\widehat{z^k_{\circ,\circ}} \leftarrow \widehat{z^k_{\circ,\circ}} + 1$;

    end for

  end for

  $\mu \leftarrow (\overline{Z}^T\overline{Z})^{-1}\overline{Z}^T Y$

  $\delta \leftarrow \frac{1}{M}(Y - \overline{Z}\mu)^T(Y - \overline{Z}\mu)$

until convergence

Calculate θ using Equation 5.8

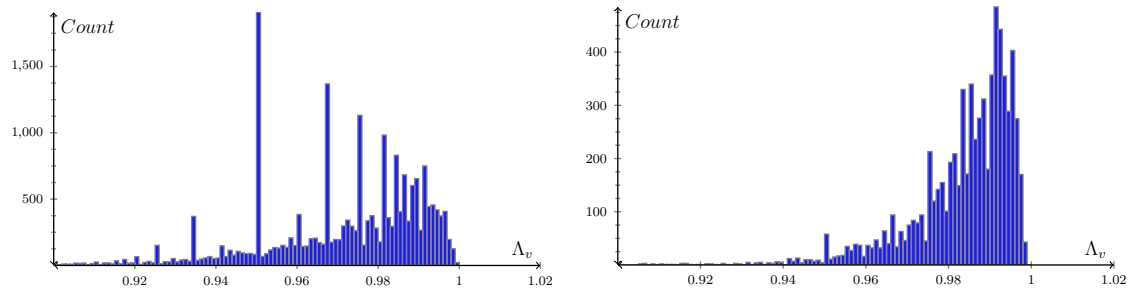
Calculate φ using Equation 5.9

return Z,θ,φ

- PubMed corpus which is a subset of articles abstracts published by PubMed. It comprises 70,287 documents with 125,652 unique terms. Total number of words in this corpus is 6,570,235 words.

- NewsAP corpus [65] which is a subset of news articles from Associated Press (AP). It has 38,500 unique terms and 453,462 words spread over 2,213 documents.

For both corpora, standard English stop words were removed. Then based on word order, Λ is calculated for all terms in the corpus as demonstrated before. Figure 5.4 shows the distribution of values of Λ_v for both corpora. In general, about 81% of NewsAP corpus terms have the value $\Lambda_v = 1$ associated to them. In the PubMed corpus the percentage of such terms is about 89%; thus these values are not displayed in the figures.



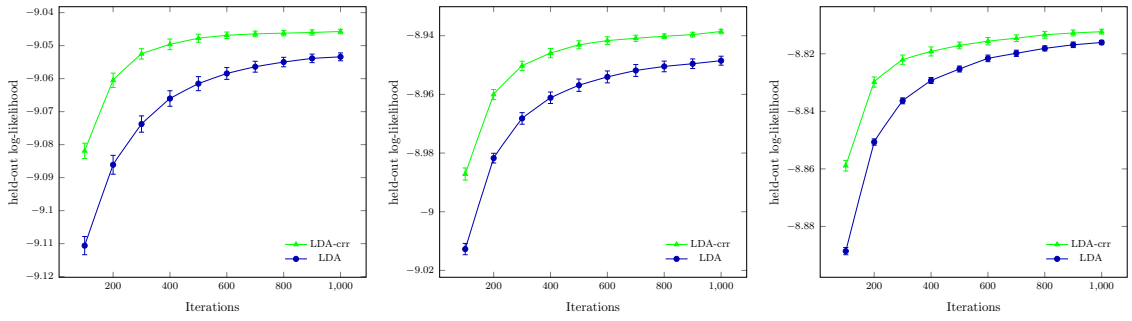
(a) PubMed (b) NewsAP

Figure 5.4: Λ_v values histogram for $v \in [1..V]$ after ignoring all $\Lambda_v = 1$.

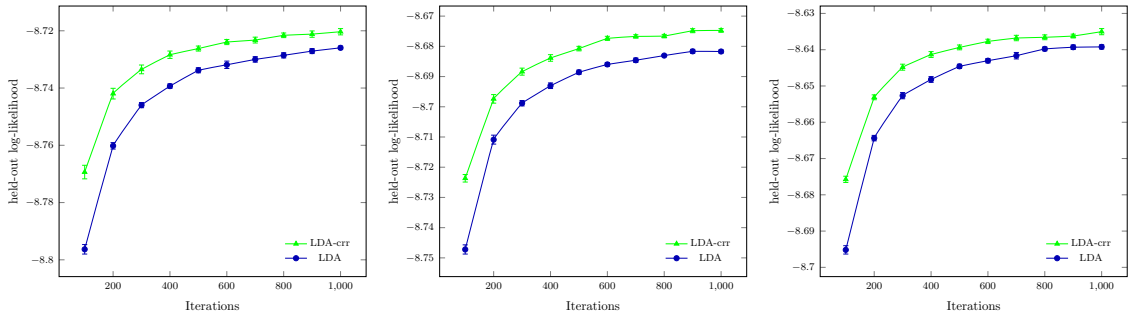
5.6.1 Perplexity Performance

An LDA-crr collapsed Gibbs sampler is implemented in Java using Algorithm 15. 50% of corpus data is used for training whereas the remaining 50% is used for testing purposes only. Firstly, the LDA-crr model is benchmarked against LDA with fixed predefined hyperparameters; this shows the advantage of incorporating word order after isolating the effect of hyperparameters. Symmetric alpha and beta are used for this setting where $\alpha_i = \frac{50}{K}$ for all $i \in [1..K]$ and $\beta_r = 0.01$ for all $r \in [1..V]$. Figure 5.5 and Figure 5.6 show held-out log-likelihood for both LDA-crr and LDA with fixed hyperparameters and different number of topics K using NewsAP and PubMed corpora respectively. LDA-crr shows a better ability to generalize to unseen documents compared with the original LDA on those two corpora. Moreover, these figures show that the LDA-crr is faster to converge than the original LDA. This is because preceding words help the model to set a better topic assignment for each word in the corpus.

Furthermore, LDA-crr performance is assessed when its hyperparameters values are learnt from input data. Consequently, LDA-crr is equipped with the GN method to learn these parameters; the resulting model is dubbed as "LDA-crrGN". Moreover, the slice sampling technique is used for the same purpose in the "LDA-crrGSS" model. LDA-GN, which is illustrated in Algorithm 13, is used as a baseline. Corpus data is split into two halves; one half is used for training whereas the other half is used for calculating held-out log-likelihood. Figure 5.7 shows perplexity scores for the three models with different numbers of topics. It shows that LDA-GN and LDA-crrGN outperform LDA-crrGSS in terms of ability to generalize to unseen documents

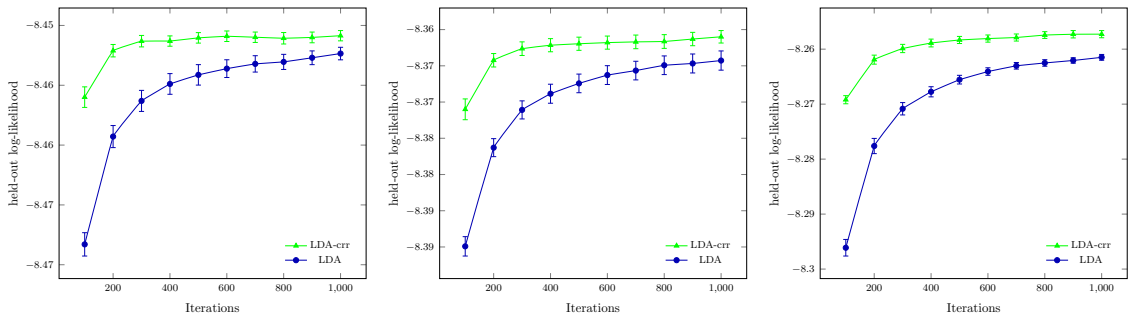


(a) $K = 5$ (b) $K = 10$ (c) $K = 25$

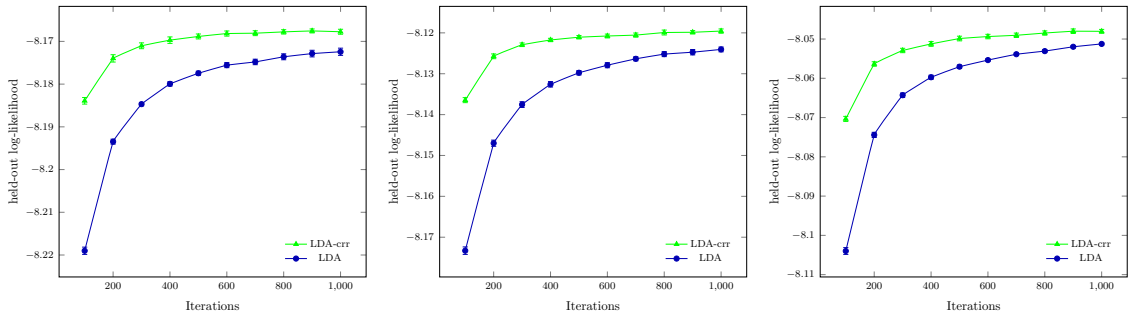


(d) $K = 50$ (e) $K = 75$ (f) $K = 150$

Figure 5.5: Held-out log-likelihood on NewsAP corpus for both LDA-crr and LDA with fixed symmetric hyperparameters settings, the higher log-likelihood the better



(a) $K = 5$ (b) $K = 10$ (c) $K = 25$



(d) $K = 50$ (e) $K = 75$ (f) $K = 150$

Figure 5.6: Held-out log-likelihood on PubMed corpus for both LDA-crr and LDA with fixed symmetric hyperparameters settings, the higher log-likelihood the better

in NewsAP corpus. Moreover, Figure 5.8 highlights the difference in performance between LDA-crrGN and LDA-GN and shows an advantage of using LDA-crrGN over LDA-GN when the number of topics gets higher. In addition, LDA-crrGN is able to converge faster than LDA-GN, as is shown by Figure 5.8 and Figure 5.9.

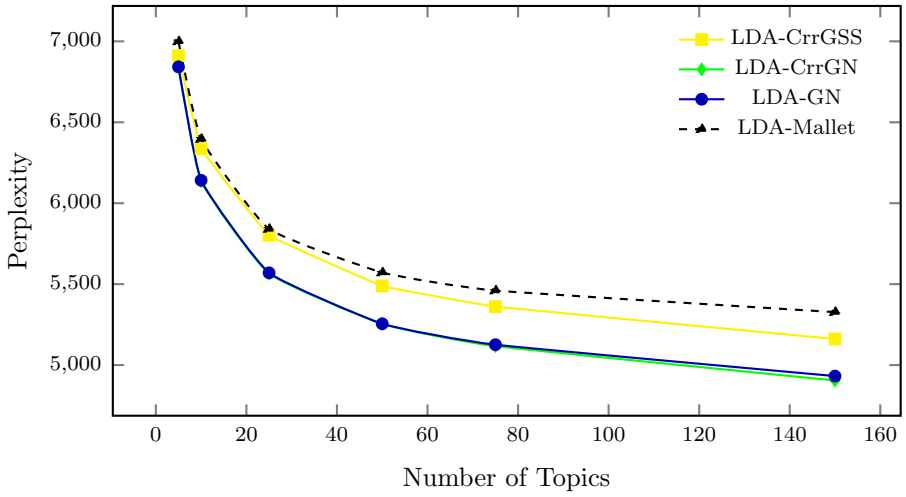
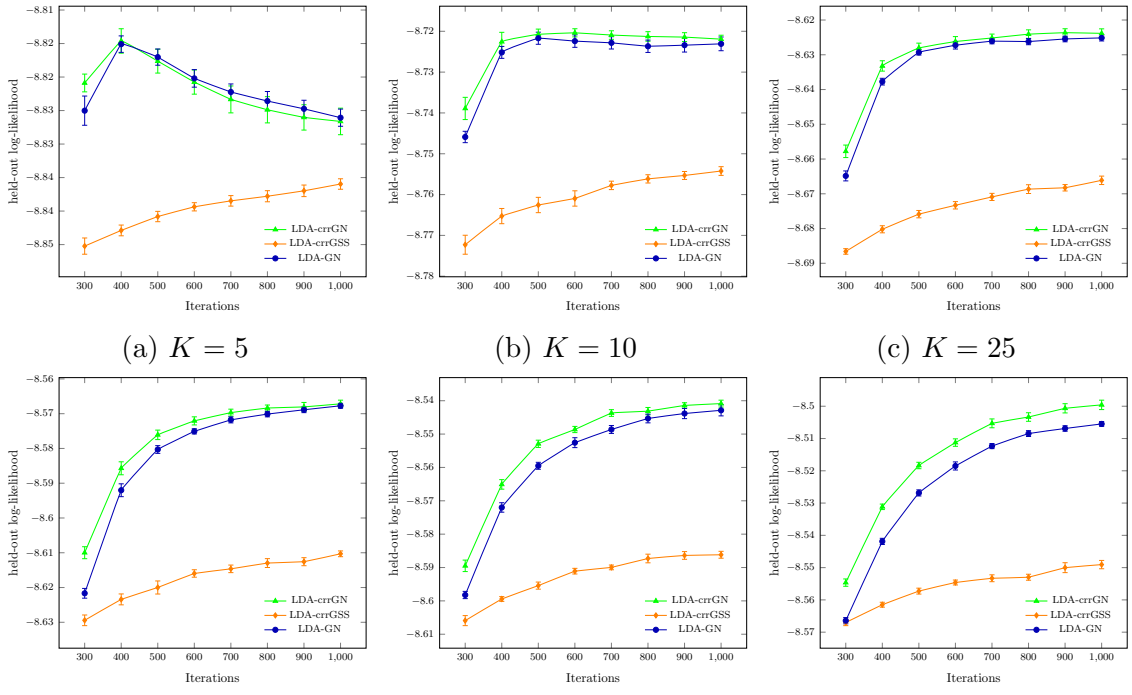


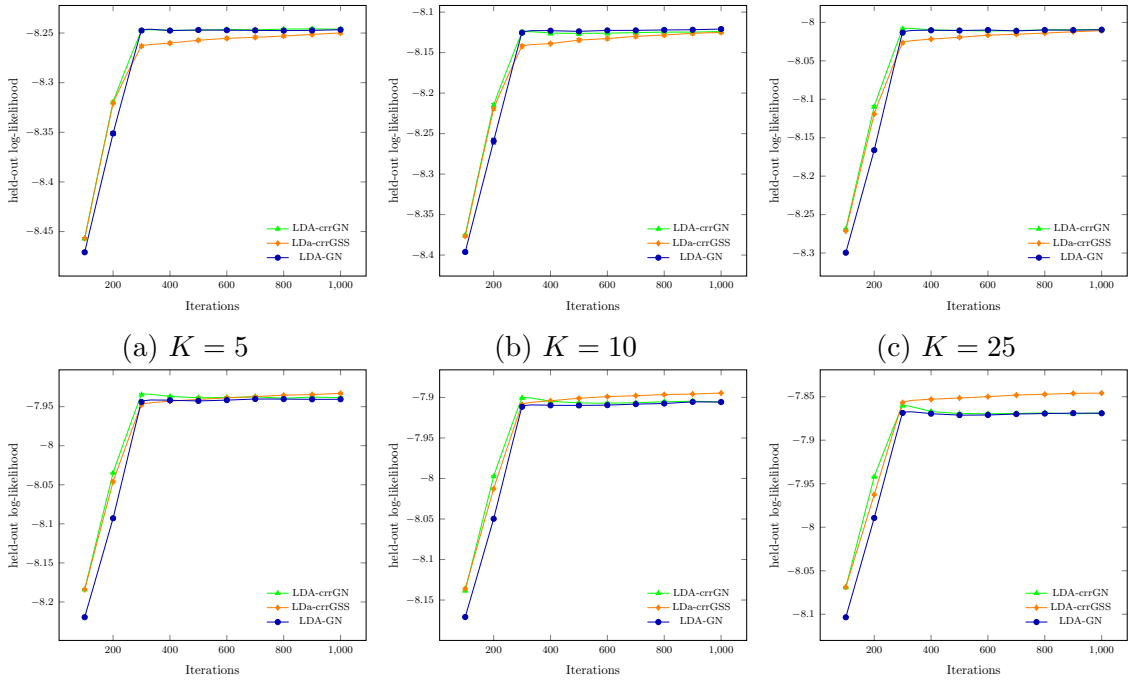
Figure 5.7: NewsAP corpus, LDA-crrGN, LDA-crrGSS, LDA-GN and LDA-Mallet Perplexity values for different number of topics



(a) $K = 5$ (b) $K = 10$ (c) $K = 25$

(d) $K = 50$ (e) $K = 75$ (f) $K = 150$

Figure 5.8: Held-out log-likelihood on NewsAP corpus for LDA-crrGN, LDA-crrGSS, and LDA-GN, the higher log-likelihood the better



(a) $K = 5$ (b) $K = 10$ (c) $K = 25$

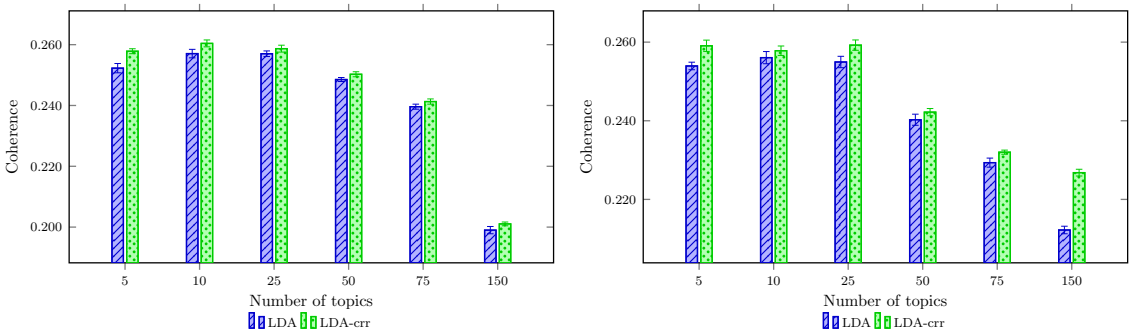
(d) $K = 50$ (e) $K = 75$ (f) $K = 150$

Figure 5.9: Held-out log-likelihood on PubMed corpus for LDA-crrGN, LDA-crrGSS, and LDA-GN, the higher log-likelihood the better

5.6.2 Coherence

In order to assess topic coherence, normalized PMI scores are calculated as described in section 2.4.2. Firstly, LDA-crr is benchmarked against LDA using fixed hyperparameters. The same settings used in the previous section are used for this experiment. After repeating the experiment ten times, coherence means and standard deviations are calculated. Table 5.1 and Table 5.2 show coherence scores for both LDA-crr and LDA using fixed symmetric hyperparameters on NewsAP and PubMed corpora respectively. Meanwhile, Figure 5.10 shows coherence scores with standard error bars for both corpora. LDA-crr shows a coherence enhancement on some topics settings. On the one hand, t-tests on PubMed corpus with settings: $K = 5$, $K = 25$, and $K = 150$ show that the difference is significant with $p < 0.05$; however, on the rest of the K settings, t-tests suggest that coherence enhancement is insignificant. On the other hand, t-tests on NewsAP corpus suggest that coherence improvement is significant when $K = 5$ with $p < 0.05$; however, coherence enhancement is insignificant for the rest of the K settings.

In addition, coherence scores are measured for LDA-crrGN, LDA-crrGSS and LDA-



(a) NewsAP (b) PubMed

Figure 5.10: Topic coherence on NewsAP and PubMed corpora for LDA and LDA-crr with fixed hyperparameters settings, the higher the better.

Table 5.1: Coherence scores for LDA-crr, LDA on NewsAP corpus.

	K=10		**K=25**		**K=50**		**K=75**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LDA-crr	0.2604	0.00358	0.2587	0.00357	0.2503	0.00260	0.2413	0.00272
LDA	0.2571	0.00442	0.2570	0.00292	0.2485	0.00207	0.2396	0.00262

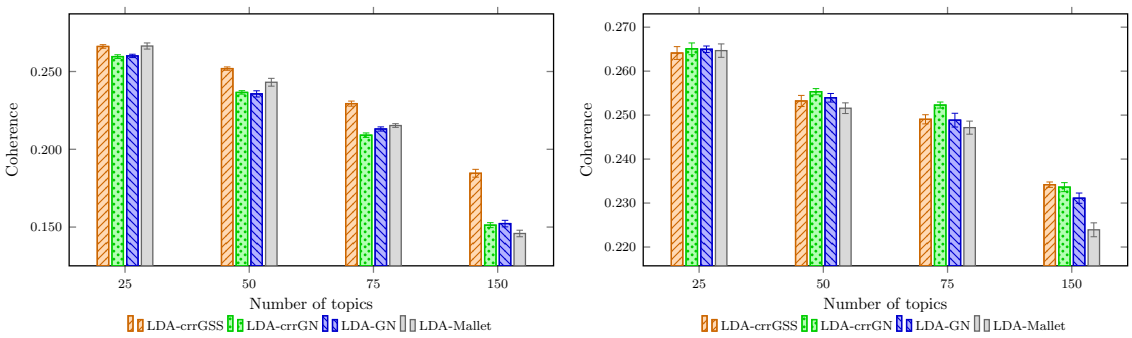
GN. The experiment is run ten times, then coherence means and standard deviations are populated in Table 5.3 and Table 5.4 and displayed with standard error bars in Figure 5.11. T-tests show that on NewsAP corpus, LDA-crrGSS outperforms both LDA-crrGN and LDA-GN in terms of coherence as the number of topics gets higher than 25 with $p < 0.05$. However, all models can discover topics with the same level of coherence on PubMed corpus.

5.7 Document Classification

This section shows the performance of LDA-crr in a classification task; where supervised versions of both LDA-GN and LDA-crrGN are used for this purpose. SLDA-GN, which is a supervised version of LDA-GN, is based on the SLDA model detailed in section 2.2.2. It uses the GN method to learn the values of hyperparameters *alpha* and *beta*. Meanwhile, SLDA-crrGN and SLDA-crrGSS, which are based on SLDA-

Table 5.2: Coherence scores for LDA-crr, LDA on PubMed corpus.

	K=10		**K=25**		**K=50**		**K=75**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LDA-crr	0.2578	0.00380	0.2592	0.00416	0.2422	0.00289	0.2320	0.00166
LDA	0.2560	0.00489	0.2549	0.00457	0.2402	0.00446	0.2293	0.00365



(a) NewsAP (b) PubMed

Figure 5.11: Topics coherence on NewsAP and PubMed corpora for LDA-GN, LDA-crrGSS, LDA-crrGN, and LDA-Mallet, the higher the better.

Table 5.3: Coherence scores for LDA-crrGN, LDA-crrGSS, LDA-GN and LDA-Mallet on NewsAP corpus.

	K=25		**K=50**		**K=75**		**K=150**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LDA-crrGN	0.2595	0.0037	0.2365	0.0035	0.2091	0.0045	0.1512	0.005
LDA-crrGSS	0.2661	0.0036	0.2518	0.0034	0.2293	0.0052	0.1846	0.0079
LDA-GN	0.2600	0.0033	0.2356	0.0063	0.2130	0.0042	0.1522	0.0066
LDA-Mallet	0.2664	0.0062	0.2431	0.0079	0.2152	0.0037	0.1458	0.0066

crr elaborated in section 5.5, use the GN method and slice sampling approaches respectively to learn their hyperparameters *alpha* and *beta*. In addition, the SLDA with an optimized symmetric *beta* and asymmetric *alpha* is used as a baseline.

5.7.1 Classification Performance

First, the classification task was performed on the Reuters corpus with ten classes. 50% of the labelled documents are used to train the models; whereas the rest of the documents are used for evaluation. The models are trained using different numbers of topics and their classification accuracy is registered for each K topics. Figure 5.12 shows the performance results with standard error displayed. The figure shows

Table 5.4: Coherence scores for LDA-crrGN, LDA-crrGSS, LDA-GN and LDA-Mallet on PubMed corpus.

	K=25		**K=50**		**K=75**		**K=150**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LDA-crrGN	0.2651	0.0042	0.2553	0.0022	0.2522	0.0021	0.2336	0.0032
LDA-crrGSS	0.2641	0.0046	0.2532	0.0039	0.2490	0.0032	0.2341	0.0019
LDA-GN	0.2649	0.0023	0.2539	0.0031	0.2488	0.0049	0.2311	0.0037
LDA-Mallet	0.2647	0.0048	0.2515	0.0038	0.2471	0.0047	0.2239	0.0050

than SLDA-crrGN has the best performance in this supervised task; where t-tests suggest that SLDA-crrGN accuracy enhancement is significant with $p < 0.01$ when it is compared with SLDA-GN on most K settings. On the other hand, SLDA with optimized asymmetric alpha and symmetric beta model is the worst performing among the tested models with $p < 0.01$ when it is compared with SLDA-crrGN for all K settings.

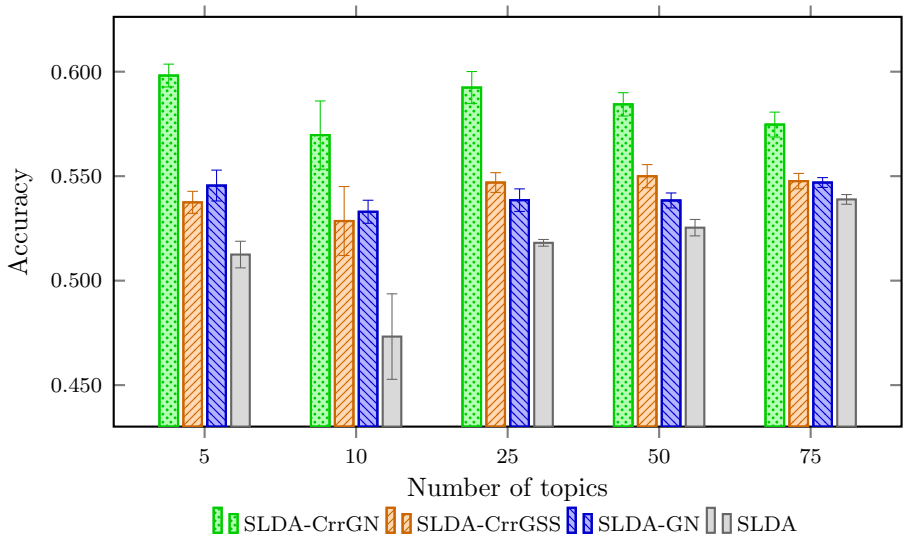


Figure 5.12: Reuters corpus, classification performance for SLDA-crrGN, SLDA-crrGSS, SLDA-GN and SLDA.

Table 5.5: Accuracy scores for SLDA-crrGN, SLDA-crrGSS, SLDA-GN and SLDA on Reuters corpus.

	K=10		**K=25**		**K=50**		**K=75**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
SLDA-crrGN	0.569	0.051	0.592	0.024	0.584	0.017	0.574	0.018
SLDA-crrGSS	0.528	0.052	0.546	0.014	0.549	0.017	0.547	0.011
SLDA-GN	0.532	0.017	0.538	0.017	0.538	0.011	0.546	0.007
SLDA	0.473	0.065	0.518	0.005	0.525	0.012	0.538	0.007

Table 5.5 shows accuracy scores for SLDA-crrGN, SLDA-crrGSS, SLDA-GN and SLDA models. Mean accuracy and standard deviation (SD) are displayed for each model and number of topics. One can see that LDA-crr is particularly useful in classification tasks compared with other models because word order information is used by the model to learn more class distinctive characteristics in training data.

5.8 Conclusions

This chapter proposed LDA-crr: a topic modelling extension which incorporates word order into the modelling process. Unlike other attempts in [149] [60], LDA-crr does not introduce high complexity to the original LDA model because the number of hidden variables is not increased significantly. This keeps the model applicable for large corpora without consuming a large amount of computational resources.

An efficient Gibbs sampler algorithm is provided for LDA-crr, which is then benchmarked against the original LDA with fixed hyperparameters. LDA-crr shows better ability to generalize to unseen documents compared with the original LDA when the same hyperparameters settings are used for both models. In terms of coherence, LDA-crr is able to learn more coherent topics especially when the number of topics gets higher; however, on the lower number of topics the coherence enhancement is insignificant.

Moreover, the GN and the GSS, which are techniques for sampling the new model's hyperparameters, are explored. In general, the GN shows a better ability to generalize to unseen documents with no observed enhancement in the topics' coherence. On the other hand, the GSS shows the same level of perplexity performance compared with both the original LDA and the LDA-GN models; however, it exhibits better topics' coherence especially when the number of topics gets higher. Consequently, one should choose the 'right' method depending on the topic modelling application. For example, for applications when topics coherence is more important such as in documents tagging in digital libraries, the GSS method can be used. Whereas, for dimensionality reduction applications, the GN technique can be the best performing one.

In addition, a supervised version of the proposed model is presented. This new extension incorporates more semantic information from the training data which leads to a better classification ability compared with the original LDA model. This is because the proposed model picks up more information from training documents which allows it to predict unseen documents' classes more accurately.

Chapter 6

Conclusions and Future Work

Topic models have many applications in machine learning, including analyzing and categorizing large data sets. Because of their unsupervised nature, topic models' evaluation is not easy; hence, model perplexity is used mainly as a performance metric. Lower perplexity value reflects a better ability to generalize to unseen documents. Moreover, topic coherence is also used as a metric, which conveys how 'close' topic words are to each other in a semantic sense. In this thesis, ideas to enhance topic modelling performance, mainly in terms of perplexity and coherence, have been explored. In the remainder of this chapter, the key findings of this thesis are summarized, and ideas for future work are proposed.

6.1 Summary of Results

In this thesis, novel topic models which discover higher quality topics in terms of perplexity and coherence are presented and evaluated as follows:

In Chapter 3, a novel multi-objective topic modelling algorithm, dubbed MOEA-TM, is presented. MOEA-TM uses the MOEA/D algorithm to optimize two objectives: a coverage objective which ensures that topics cover all corpus documents, and a PMI objective which is responsible for enhancing topic coherence. MOEA-TM can perform better than standard LDA in terms of coherence; however, when it is initialized using an LDA model, MOEA-TM was not able to optimize the perplexity measure. Later, a genetic algorithm was designed with an objective to optimize the

perplexity of the LDA model. This is a novel GA which has the LDA model's log-likelihood as a fitness function. Although it can optimize the model's log-likelihood by up to ten percent, the perplexity scores are not optimized; in fact, perplexity values deteriorate as the number of topics gets higher. This shows that the pure optimization technique is able to enhance topics coherence but not the ability to generalize to a held-out unseen document. On the one hand, coherence is associated with top words chosen for each of the inferred topics. Thus, optimization is successful in this mission as it directly targets topics' top words. This is useful in some topic modelling applications which need mainly the topics' top words such as documents tagging. On the other hand, perplexity, which indicates the model's ability to generalize to unseen documents, is affected mainly by how well the estimated model represents the true posterior given observed data. Optimization tends to find the mode, which might not be well representing especially in multi-modal distributions. Other techniques such as MCMC focus on the expected value which provides better representation; thus, output models tend to have a higher ability to predict held-out documents. Other techniques are investigated later to enhance perplexity.

Experimenting with the LDA model shows that its hyperparameters play a significant role in the quality of the output topics. A closer look at the LDA model reveals that it comprises two multivariate Pólya distributions combined. Consequently, finding faster and better methods for learning multivariate Pólya distribution's parameters leads to higher quality topic models. Thus, in Chapter 4 the GN and the GSS techniques to learn the parameters of a multivariate Pólya distribution are proposed. The GN method uses numerical optimization whereas the GSS is based on slice sampling [112]. The new techniques can provide accurate results faster compared with the state-of-art methods available in the literature. Both techniques are also able to achieve lower perplexity scores when benchmarked against the original LDA. Moreover, asymmetric settings are used for the proposed models' hyperparameters α and β which provides more flexibility for the words to be distributed in topics. Also, the performance of these models in a supervised classification task is measured. Two approaches were adopted to achieve that: firstly, both techniques are used in the

standard supervised topic model SLDA and benchmarked against the SLDA in its original priors settings. Secondly, MC-LDA is used in a spam filtering task, which shows that the GN method is less sensitive to the threshold setting compared with the original LDA. This is because the model equipped with the GN method reflects a better representation which leads to a better discrimination of the classes. Thus, the work in this chapter shows how working on better LDA priors may enhance the topic model's quality and improve its ability to generalize. Moreover, Chapter 4 shows that the sampling technique for estimating topic models' hyperparameters is less successful than an optimization technique. The reason behind this might be that providing a stable estimation for these parameters could be better than sampling different values during the inference process. Next, more work is done in the way words are modelled which can be done by incorporating more info from text such as word order.

In Chapter 5, a novel model LDA-crr is proposed. Compared with the original LDA model, which ignores word order information, LDA-crr incorporates word order in the modelling process, and it only introduces minor additional complexity to the original model. In general, LDA-crr converges faster than LDA using both fixed concentration parameters and dynamic ones. With fixed parameters in use, LDA-crr is not only able to converge faster than the LDA but also scores lower perplexity values which lead to a better ability to generalize to unseen documents. Generally speaking, using dynamic parameters, the LDA-crr equipped with the GN approach to learn concentration parameters shows the best performance in terms of perplexity. Meanwhile, the LDA-crr combined with a slice sampling technique generally shows the best performance in terms of coherence. Consequently, incorporating word order helps in producing quality models with higher ability to generalize to unseen documents. This comes with an only small increase in the complexity of the model compared with earlier attempts in the literature [149] [60]. Lower perplexity usually indicates higher classification accuracy; thus, to measure the LDA-crr performance on a supervised task, SLDA was extended to incorporate word order. As a result, SLDA-crr is developed and benchmarked against the SLDA-GN and the original

SLDA. Classification performance on a ten classes corpus shows that SLDA-crr has the best accuracy when it is used with the GN method. This correlates well with the perplexity results and clearly shows that the proposed model indeed can pick up more valuable information from corpus documents which allow it to predict held-out documents' classes more accurately.

6.2 Future Research Work

This section highlights some future work directions that this research may lead to. This includes enhancing the execution speed of LDA-crr and incorporating word order in LDA extensions and other interesting topic models.

6.2.1 Sparse Models

One can see a Gibbs sampling implementation as a repetitive task of sampling each hidden variable given other variables until convergence. In the case of the LDA, a collapsed Gibbs sampler samples a topic assignment for each word. Consequently, a more efficient technique to sample a topic assignment would reduce the amount of time taken until convergence. Real-world topic models are highly sparse, especially when the number of topics gets higher. Thus, SparseLDA [160], which introduces a more efficient sampling technique, is able to reduce the amount of resources significantly. Models which are proposed in this thesis can be implemented using the efficient technique available in [160]. SparseLDA enables these models to handle large corpora more efficiently.

6.2.2 Informative Priors

One potential area of future work for LDA-GN is to investigate the placement of informed priors before the *alpha* and *beta* variables. Such may be available for many applications (including, for example, updating a topic model following an extension of the corpus). Meanwhile, the quality of the topic models learnt by LDA-GN seems to augur well for their use in supervised learning tasks; spam classification is one

example, but other tasks in the general area of supervised document classification may benefit from LDA-GN in the context of the SLDA-GN and MC-LDA approach. This may be especially fruitful in the case of discrimination tasks that involve 'close' categories (e.g. 'finance' vs 'insurance').

6.2.3 LDA Extensions

LDA has many extensions in the literature which includes: Author Topic Model [131], Labelled LDA [127], and hierarchical topic models [89] [58] [16]. Most of these extensions use the 'Bag of Words' assumption which ignores word order. In this thesis, the LDA model is enhanced without the need for using external information, which opens the door for incorporating changes to any other model built on LDA easily. It is particularly tempting to experiment how incorporating word order into Correlated Topic Model (CTM) [16] may enhance performance; in CTM, topics are not independent, and word order may play a larger role in building a higher quality topic model.

6.2.4 Other Applications

This thesis targeted topic modelling in the context of textual data. In this context, there are many applications [21] which can benefit directly from the ideas proposed in this work. Such applications include: historical documents, understanding scientific publications, computational social science, fiction and literature. Also, topic modelling has many applications beyond the context of textual data. Thus, it is tempting to investigate relaxing the 'Bag of Words' assumption on applications such as image classification and annotation [28] [48]. The first step in image annotation is to represent the image as a bag of visual words to convert the image from continuous to discrete space. This segmentation can be done using a grid or using a feature extraction algorithm. Feature spatial information might be relevant in images as well as in text [92]; where features contribute to defining topics differently. The effect of feature sequence is as yet little studied in the literature; however, might hold high potential in enhancing the accuracy. For example, some features may appear solely

in few image topics; whereas, others may appear in many topics. Those features that appear in few topics hold more semantic importance and they can be used to learn more about the topics of nearby features which are likely to share the same topics.

The same principle applies to use topic modelling in analyzing musical data [71]. In the context of sound processing, Mel-frequency cepstral coefficients (MFCC) technique [121] can be used to represent the sound into features. Again, using 'Bag of Words' representation lose vital information that might help to estimate higher quality models.

References

[1] B. Adams, L. Bauman, W. Bohnhoff, K. Dalbey, M. Ebeida, J. Eddy, M. Eldred, P. Hough, K. Hu, J. Jakeman, J. Stephens, L. Swiler, D. Vigil, and T. Wildey. *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.7 Users Manual.* Sandia Technical Report SAND2014-4633, November 2017.

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Combining stochastic block models and mixed membership for statistical network analysis. In E. Airoldi, D. M. Blei, S. E. Fienberg, A. Goldenberg, E. P. Xing, and A. X. Zheng, editors, *Statistical Network Analysis: Models, Issues, and New Directions*, Lecture Notes in Computer Science, pages 57–74. Springer Berlin Heidelberg, 2007.

[3] J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 139–177, 1982.

[4] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 37–45. ACM, 1998.

[5] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami. Contributions to the study of SMS spam filtering: New collection and results. In *Proceedings of the 11th ACM Symposium on Document Engineering*, DocEng '11, pages 259–262. ACM, 2011.

[6] A. Anandkumar, Y. kai Liu, D. J. Hsu, D. P. Foster, and S. M. Kakade. A spectral algorithm for latent dirichlet allocation. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 917–925. Curran Associates, Inc., 2012.

[7] S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 280–288. JMLR: W\&CP 28 (2), 2013.

[8] S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization – provably. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 145–162, New York, NY, USA, 2012. ACM.

[9] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press.

[10] O. B. Augusto, F. Bennis, and S. Caro. A new method for decision making in multi-objective optimization problems. *Pesquisa Operacional*, 32:331 – 369, 08 2012.

[11] S. Bao, S. Xu, L. Zhang, R. Yan, Z. Su, D. Han, and Y. Yu. Joint emotion-topic modeling for social affective text mining. In *2009 Ninth IEEE International Conference on Data Mining*, pages 699–704, Dec 2009.

[12] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *J. Mach. Learn. Res.*, 3:1107–1135, Mar. 2003.

[13] H. P. Benson. An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *Journal of Global Optimization*, 13(1):1–24, 1998.

[14] I. Bíró. *Document Classification with Latent Dirichlet Allocation*. PhD thesis, Etvs Lornd University, Faculty of Informatics, 2009.

[15] I. Bíró, J. Szabó, and A. A. Benczúr. Latent Dirichlet allocation in web spam filtering. In *Adversarial Information Retrieval on the Web*, pages 29–32, 2008.

[16] D. M. Blei and J. D. Lafferty. A correlated topic model of science. *Ann. Appl. Stat.*, 1(1):17–35, June 2007.

[17] D. M. Blei and J. D. Mcauliffe. Supervised topic models. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 121–128. Curran Associates, Inc., 2008.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[19] M. Borda. *Fundamentals in Information Theory and Coding*. Springer Berlin Heidelberg, 2011.

[20] G. Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40, 2009.

[21] J. Boyd-Graber, Y. Hu, and D. Mimno. Applications of topic models. *Foundations and Trends in Information Retrieval*, 11(2-3):143–296, 2017.

[22] W. Buntine. Estimating likelihoods for topic models. In Z.-H. Zhou and T. Washio, editors, *Advances in Machine Learning*, volume 5828 of *Lecture Notes in Computer Science*, pages 51–64. Springer Berlin Heidelberg, 2009.

[23] W. Buntine and A. Jakulin. Applying discrete pca in data analysis. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 59–66. AUAI Press, 2004.

[24] A. J.-B. Chaney and D. M. Blei. Visualizing topic models. In *ICWSM*, 2012.

[25] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-graber, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans,

J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc., 2009.

[26] X. Chen, X. Hu, X. Shen, and G. Rosen. Probabilistic topic modeling for genomic data interpretation. In *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pages 149–152, Dec 2010.

[27] S. Chib. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90(432):1313–1321, 1995.

[28] W. Chong, D. Blei, and L. Fei-Fei. Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1903–1910. IEEE, 2009.

[29] J. Chuang, C. D. Manning, and J. Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 74–77. ACM, 2012.

[30] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, Mar. 1990.

[31] C. Coello, G. Lamont, and D. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer US, 2007.

[32] C. A. Coello Coello. Evolutionary multi-objective optimization: A historical view of the field. *Comp. Intell. Mag.*, 1(1):28–36, Nov. 2006.

[33] C. A. Coello Coello. *Evolutionary Multi-Objective Optimization: Basic Concepts and Some Applications in Pattern Recognition*, pages 22–33. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[34] R. Cohen, I. Aviram, M. Elhadad, and N. Elhadad. Redundancy-aware topic modeling for patient record notes. *PLoS One*, 9(2), Feb 2014.

[35] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[36] P. J. Davis. *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*, chapter Gamma Function and Related Functions. Dover Publications Inc., 1972.

[37] K. Deb and D. Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001.

[38] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[39] P. Del Moral, A. Doucet, and A. Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[41] J. G. Dias and M. Wedel. An empirical comparison of em, sem and mcmc performance for problematic gaussian mixture likelihoods. *Statistics and Computing*, 14(4):323–332, Oct. 2004.

[42] J. M. Dickey. Multiple hypergeometric functions: Probabilistic interpretations and statistical uses. *Journal of the American Statistical Association*, 78(383):628–637, 1983.

[43] C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Comput. Stat. Data Anal.*, 52(8):3913–3927, Apr. 2008.

[44] M. Dishon and G. H. Weiss. Small sample comparison of estimation methods for the beta distribution. *Journal of Statistical Computation and Simulation*, 11(1):1–11, 1980.

[45] G. Doyle and C. Elkan. Financial Topic Models. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*, Whistler, Canada, 2009.

[46] E. A. Erosheva, S. E. Fienberg, and C. Joutard. Describing disability through individual-level mixture models for multivariate binary data. *The Annals of Applied Statistics*, 1(2):502–537, 12 2007.

[47] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[48] Y. Feng and M. Lapata. Topic models for image annotation and text illustration. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 831–839, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[49] C. J. Fillmore and B. T. Atkins. Describing polysemy: The case of 'crawl'. In Y. Ravin and C. Leacock, editors, *Polysemy: Theoretical and Computational Approaches*. Oxford University Press, 2000.

[50] P. C. Fishburn. Letter to the editoradditive utilities with incomplete product sets: Application to priorities and assignments. *Operations Research*, 15(3):537–542, 1967.

[51] K. R. Fowler, J. P. Reese, C. E. Kees, J. E. Dennis, Jr., C. T. Kelley, C. T. Miller, C. Audet, A. J. Booker, G. Couture, R. W. Darwin, M. W. Farthing, D. E. Finkel, J. M. Gablonsky, G. Gray, and T. G. Kolda. A comparison of derivative-free optimization methods for groundwater supply and hydraulic

capture community problems. *Advances in Water Resources*, 31(5):743–757, May 2008.

[52] C. W. Fox and S. J. Roberts. A tutorial on variational bayesian inference. *Artificial Intelligence Review*, 38(2):85–95, 2012.

[53] S. Gao and H. Li. Octave-dependent probabilistic latent semantic analysis to chorus detection of popular song. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 979–982, New York, NY, USA, 2015. ACM.

[54] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, Nov. 1984.

[55] G. K. Gerber, R. D. Dowell, T. S. Jaakkola, and D. K. Gifford. Automated discovery of functional generality of human gene expression programs. *PLoS Comput Biol*, 3(8):1–15, 08 2007.

[56] G. A. Gray and T. G. Kolda. Algorithm 856: Appspack 4.0: Asynchronous parallel pattern search for derivative-free optimization. *ACM Trans. Math. Softw.*, 32(3):485–507, Sept. 2006.

[57] G. A. Gray, T. G. Kolda, K. L. Sale, and M. M. Young. Optimizing an empirical scoring function for transmembrane protein structure determination. *INFORMS Journal on Computing*, 16(4):406–418, 2004.

[58] D. Griffiths and M. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16:17, 2004.

[59] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of The National Academy of Sciences*, 101:5228–5235, Apr. 2004.

[60] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum. Integrating topics and syntax. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press, 2005.

[61] D. Hadka. Moea framework: A free and open source java framework for multiobjective optimization, 2014.

[62] Y. Y. Haimes, L. Ladson, and D. A. Wismer. Bicriterion formulation of problems of integrated system identification and system optimization. *IEEE Transactions on Systems Man and Cybernetics*, (3):296, 1971.

[63] D. Hall, D. Jurafsky, and C. D. Manning. Studying the history of ideas using topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 363–371, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[64] E. Halperin and R. M. Karp. Automata, languages and programming: Algorithms and complexity (icalp-a 2004) the minimum-entropy set cover problem. *Theoretical Computer Science*, 348(2):240 – 250, 2005.

[65] D. Harman. Overview of the first text retrieval conference (trec-1). In *Proceedings of the First Text Retrieval Conference (TREC-1)*, pages 1–20, 1992.

[66] Z. S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.

[67] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *J. Mach. Learn. Res.*, 14:1303–1347, May 2013.

[68] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57. ACM, 1999.

[69] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *J. Mach. Learn.*, 42(1-2):177–196, Jan. 2001.

[70] P. D. Hough, T. G. Kolda, and V. J. Torczon. Asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Scientific Computing*, 23(1):134–156, 2001.

[71] D. Hu and L. K. Saul. A probabilistic topic model for unsupervised learning of musical key-profiles. In *ISMIR*, pages 441–446, 2009.

[72] F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *Proceedings of the 31th International Conference on Machine Learning*, ICML 2014, pages 754–762. JMLR.org, June 2014.

[73] C. Hwang and A. Masud. *Multiple objective decision making, methods and applications: a state-of-the-art survey*. Lecture notes in economics and mathematical systems. Springer-Verlag, 1979.

[74] A. Jaszkiewicz and J. Branke. *Interactive Multiobjective Evolutionary Algorithms*, pages 179–193. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[75] Y. S. Jeong, W. J. Lee, and H. J. Choi. Topic grouping by spectral clustering. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, pages 657–661, Feb 2014.

[76] O. Khalifa, D. W. Corne, M. Chantler, and F. Halley. Multi-objective topic modeling. In R. Purshouse, P. Fleming, C. Fonseca, S. Greco, and J. Shaw, editors, *Evolutionary Multi-Criterion Optimization*, volume 7811 of *Lecture Notes in Computer Science*, pages 51–65. Springer Berlin Heidelberg, 2013.

[77] S. Kim, P. Georgiou, and S. Narayanan. Latent acoustic topic models for unstructured audio classification. *APSIPA Transactions on Signal and Information Processing*, 1:e6, 2012.

[78] J. D. Knowles, R. A. Watson, and D. W. Corne. *Reducing Local Optima in Single-Objective Problems by Multi-objectivization*, pages 269–283. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[79] T. G. Kolda. Revisiting asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal on Optimization*, 16(2):563–586, 2005.

[80] T. G. Kolda and V. J. Torczon. *Understanding Asynchronous Parallel Pattern Search*, pages 323–342. Springer US, Boston, MA, 2003.

[81] T. G. Kolda and V. J. Torczon. On the convergence of asynchronous parallel pattern search. *SIAM Journal on Optimization*, 14(4):939–964, 2004.

[82] D. Kuang, J. Choo, and H. Park. *Nonnegative Matrix Factorization for Interactive Topic Modeling and Document Clustering*, pages 215–243. Springer International Publishing, Cham, 2015.

[83] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955.

[84] H. W. Kuhn. Variants of the hungarian method for assignment problems. *Naval Research Logistics Quarterly*, 3(4):253–258, 1956.

[85] S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

[86] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.

[87] M. Lee, Z. Liu, R. Kelly, and W. Tong. Of text and gene – using text mining methods to uncover hidden knowledge in toxicogenomics. *BMC Systems Biology*, 8(1):1–11, 2014.

[88] S. Leeds and A. E. Gelfand. Estimation for Dirichlet mixed models. *Naval Research Logistics (NRL)*, 36(2):197–214, 1989.

[89] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 577–584, New York, NY, USA, 2006. ACM.

[90] J. Liang and Y. Q. Chen. Optimization of a fed-batch fermentation process control competition problem using the neos server. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 217(5):427–432, 2003.

[91] J. S. Liu. The collapsed Gibbs sampler in Bayesian computations with application to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, 1994.

[92] A. P. López-Monroy, M. Montes-y Gómez, H. J. Escalante, A. Cruz-Roa, and F. A. González. Improving the bovw via discriminative visual n-grams and mkl strategies. *Neurocomput.*, 175(PA):768–781, Jan. 2016.

[93] W. Luo, B. Stenger, X. Zhao, and T.-K. Kim. Automatic topic discovery for multi-object tracking. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3820–3826. AAAI Press, 2015.

[94] D. J. C. MacKay and L. C. B. Peto. A hierarchical Dirichlet language model. *Journal of Natural Language Engineering*, 1:289–308, 9 1995.

[95] T. Malisiewicz, J. C. Huang, and A. A. Efros. Detecting objects via multiple segmentations and latent topic models, 2006.

[96] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[97] A. Mardani, A. Jusoh, K. M. Nor, Z. Khalifah, N. Zakwan, and A. Valipour. Multiple criteria decision-making techniques and their applications a review of the literature from 2000 to 2014. *Economic Research-Ekonomska Istraivanja*, 28(1):516–571, 2015.

[98] A. K. McCallum. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[99] P. McCullagh and J. Nelder. *Generalized Linear Models, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1989.

[100] A. Messac, A. Ismail-Yahaya, and C. Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and Multidisciplinary Optimization*, 25(2):86–98, 2003.

[101] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[102] V. Metsis, I. Androutsopoulos, and G. Paliouras. Spam filtering with naive Bayes - which naive Bayes? In *Third Conference on Email and Anti-Spam (CEAS)*, 2006.

[103] K. Miettinen. *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. Springer US, 1999.

[104] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[105] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, pages 352–359, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[106] T. P. Minka. Estimating a Dirichlet distribution. Technical report, Microsoft, 2000.

[107] T. P. Minka. Bayesian inference, entropy, and the multinomial distribution. Technical report, 2003.

[108] I. Mukherjee and D. M. Blei. Relative performance guarantees for approximate inference in latent dirichlet allocation. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1129–1136. Curran Associates, Inc., 2009.

[109] J. Murdock and C. Allen. Visualization techniques for topic model checking. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 4284–4285. AAAI Press, 2015.

[110] G. M. Namata, P. Sen, M. Bilgic, and L. Getoor. Collective classification for text classification. In M. Sahami and A. Srivastava, editors, *Text Mining: Classification, Clustering, and Applications*. Taylor and Francis Group, 2009.

[111] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[112] R. M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 06 2003.

[113] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972.

[114] D. Newman, E. V. Bonilla, and W. Buntine. Improving topic coherence with regularized topic models. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 496–504. Curran Associates, Inc., 2011.

[115] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 100–108, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[116] D. Newman, Y. Noh, E. Talley, S. Karimi, and T. Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, JCDL '10, pages 215–224, New York, NY, USA, 2010. ACM.

[117] M. A. Newton and A. E. Raftery. Approximate bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):3–48, 1994.

[118] V. A. Nguyen, J. Boyd-Graber, and P. Resnik. Sometimes average is best: The importance of averaging for prediction using mcmc inference in topic modeling. In *Empirical Methods in Natural Language Processing*, 2014.

[119] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11(122-129):1–2, 2010.

[120] N. L. of Medicine NLM. PubMed, 2016. Available at `https://www.nlm.nih.gov/databases/download/pubmed_medline.html`.

[121] C. K. On, P. M. Pandiyan, S. Yaacob, and A. Saudi. Mel-frequency cepstral coefficient analysis in speech recognition. In *2006 International Conference on Computing Informatics*, pages 1–5, June 2006.

[122] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 271. Association for Computational Linguistics, 2004.

[123] V. Pareto. Cours d'Economie politique. *Revue Economique*, 7, 1897.

[124] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 569–577, New York, NY, USA, 2008. ACM.

[125] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

[126] H. Raïffa and R. Schlaifer. *Applied statistical decision theory.* Studies in managerial economics. Division of Research, Graduate School of Business Adminitration, Harvard University, 1961.

[127] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[128] P. Resnik, W. Armstrong, L. Claudino, T. Nguyen, V.-A. Nguyen, and J. Boyd-Graber. Beyond lda: exploring supervised topic modeling for depression-related language in twitter. *NAACL HLT 2015*, page 99, 2015.

[129] M. Roberts, B. Stewart, and D. Tingley. *Navigating the Local Modes of Big Data: The Case of Topic Models.* Cambridge University Press, New York, 2016.

[130] G. Ronning. Maximum likelihood estimation of Dirichlet distributions. *Journal of Statistical Computation and Simulation*, 32(4):215–221, 1989.

[131] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[132] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, pages 1605–1614. IEEE Computer Society, 2006.

[133] T. Rydn. Em versus markov chain monte carlo for estimation of hidden markov models: a computational perspective. *Bayesian Anal.*, 3(4):659–688, 12 2008.

[134] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860. ACM, 2010.

[135] G. Sakkis, I. Androutsopoulos, and C. D. Spyropoulos. A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval*, 6:49–73, 2003.

[136] T. Salimans, D. P. Kingma, and M. Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *ICML*, 2015.

[137] Y. Shen, C. Archambeau, D. Cornford, M. Opper, J. Shawe-Taylor, and R. Barillec. A comparison of variational and markov chain monte carlo methods for inference in partially observed stochastic dynamic systems. *Journal of Signal Processing Systems*, 61(1):51–59, 2010.

[138] J. Snyder, R. Knowles, M. Dredze, M. Gormley, and T. Wolfe. Topic models and metadata for visualizing text corpora. In *Proceedings of the 2013 NAACL HLT Demonstration Session*, pages 5–9, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[139] D. Sontag and D. Roy. Complexity of inference in latent dirichlet allocation. In *NIPS*, 2011.

[140] D. Sontag and D. M. Roy. Complexity of Inference in Topic Models. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*, Whistler, Canada, 2009.

[141] R. E. Steuer and E.-U. Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344, 1983.

[142] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and*

Computational Natural Language Learning, EMNLP-CoNLL '12, pages 952–961, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[143] M. Steyvers and T. Griffiths. *Handbook of Latent Semantic Analysis*, chapter Probabilistic Topic Models, pages 427–448. University of Colorado Institute of Cognitive Science Series. Lawrence Erlbaum Associates, 2007.

[144] M. Steyvers and T. L. Griffiths. Rational analysis as a link between human memory and information retrieval. In N. Chater and M. Oaksford, editors, *The Probabilistic Mind: Prospects for Bayesian Cognitive Science*. Oxford University Press, 2008.

[145] Q. Su, K. Xiang, H. Wang, B. Sun, and S. Yu. *Using Pointwise Mutual Information to Identify Implicit Features in Customer Reviews*, pages 22–30. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[146] E. M. Talley, D. Newman, D. Mimno, B. W. Herr, H. M. Wallach, G. A. P. C. Burns, A. G. M. Leenders, and A. McCallum. Database of nih grants using machine-learned categories and graphical clustering. *Nat Meth*, 8(6):443–444, Jun 2011.

[147] Y. W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent Dirichlet allocation. In *In NIPS 19*, pages 1353–1360, 2007.

[148] C. von Lücken, B. Barán, and C. Brizuela. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 58(3):707–756, 2014.

[149] H. M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 977–984, New York, NY, USA, 2006. ACM.

[150] H. M. Wallach. *Structured Topic Models for Language*. PhD thesis, University of Cambridge, 2008.

[151] H. M. Wallach, D. Mimno, and A. McCallum. Rethinking lda: Why priors matter. In *Proceedings of Neural Information Processing Systems*, NIPS '22, pages 1973–1981, 2009.

[152] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112. ACM, 2009.

[153] X. Wang and A. McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 424–433. ACM, 2006.

[154] Y. Wang and J. Zhu. Spectral methods for supervised topic models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1511–1519. Curran Associates, Inc., 2014.

[155] S. Watanabe and K. Sakakibara. Multi-objective approaches in a single-objective optimization environment. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1714–1721 Vol. 2, Sept 2005.

[156] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM.

[157] N. Wicker, J. Muller, R. K. R. Kalathur, and O. Poch. A maximum likelihood approximation method for Dirichlet's parameter estimation. *Computational Statistics and Data Analysis*, 52:1315–1322, 2008.

[158] P. Xie, D. Yang, and E. P. Xing. Incorporating word correlation knowledge into topic modeling. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, 2015.

[159] Y. Yang, D. Downey, and J. Boyd-Graber. Efficient methods for incorporating knowledge into topic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 308–317, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[160] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 937–946, New York, NY, USA, 2009. ACM.

[161] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, Dec 2007.

[162] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32 – 49, 2011.

[163] J. Zhu, A. Ahmed, and E. P. Xing. Medlda: Maximum margin supervised topic models. *J. Mach. Learn. Res.*, 13(1):2237–2278, Aug. 2012.